UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE ESTADÍSTICAS

MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DEL NEGOCIO

TRABAJO FIN DE MÁSTER

ANÁLISIS DEL SECTOR CINEMATOGRÁFICO

Selección, extracción, creación y análisis de variables provenientes de diversas fuentes web, como aporte inicial a la minería de datos.

Fortunato Taronna Pumilia

Cotutores:

Dr. Antonio Pareja Lora

Dr. Javier Portela García-Miguel

2015

AGRADECIMIENTOS.

Me gustaría aprovechar este espacio para mostrar mi más profundo agradecimiento a Jose Jaime Díaz García, sin cuya valiosa ayuda todo el proceso habría sido mucho más trabajoso. Sin lugar a dudas este trabajo también es tuyo.

Contenido

Índice de Tablas

Índice de Imágenes

1 INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

En los últimos años ha habido una explosión en la cantidad de datos disponibles; a las fuentes tradicionales, se han sumado otras fuentes, como servicios web, registros de transacciones en línea, datos gubernamentales abiertos, datos originados por sensores, datos generados por redes sociales (Twitter, YouTube, Facebook, etc.). Ahora el problema no es encontrar los datos, sino determinar cómo pueden utilizarse para nuestro beneficio.

Como parte inicial y fundamental de este problema nos encontramos con la fase de recopilación de datos. Diversas fuentes, presentados con estructuras muy variadas y en grandes cantidades puede ser un factor agobiante a la hora de iniciar un proyecto de análisis de información y generación de conocimientos. Los datos existen, pero ¿cómo podemos obtenerlos?, y no se trata de utilizar solamente una fuente de datos propia, sino que está creciendo la utilización de mezclas de distintas fuentes de datos.

Estos nuevos retos requieren nuevas habilidades, no necesarias hasta ahora en la generación de conocimientos tradicional. La capacidad de poder compilar una serie de datos relacionados, utilizando diversas fuentes, es parte primordial de un perfil profesional surgido en los últimos años: el del científico de datos.

Dentro de este auge de generación de datos, la industria cinematográfica se encuentra a la cabeza, con productoras y estudios utilizando cantidades considerables de recursos para posicionar sus productos, y un número de sitios web especializados y destinados a saciar el interés público por dicha industria, donde sus consumidores comparten un sinfín de información por medio de sus redes sociales. Estos aspectos hacen de este sector un objetivo atractivo e interesante de estudio.

Es por esta razón por la que decidimos proponer la creación de un compendio de datos de este sector, proveniente de múltiples fuentes; más específicamente, una base de datos compuesta de información originaria de estos sitios especializados mencionados anteriormente. Esta base de datos será completada posteriormente con información

pudiéramos extraer de redes sociales como Twitter y YouTube, mediante el uso y explotación de herramientas y técnicas informáticas.

La minería de datos se ocupa de proponer técnicas y herramientas para explorar y analizar grandes cantidades de datos. Estas deben cumplir con 3 características básicas: velocidad de procesamiento, volumen alto de datos y variedad de la información obtenida. Por ello, la creación de bases de datos amplias y confiables siempre ha sido uno de los objetivos de esta disciplina.

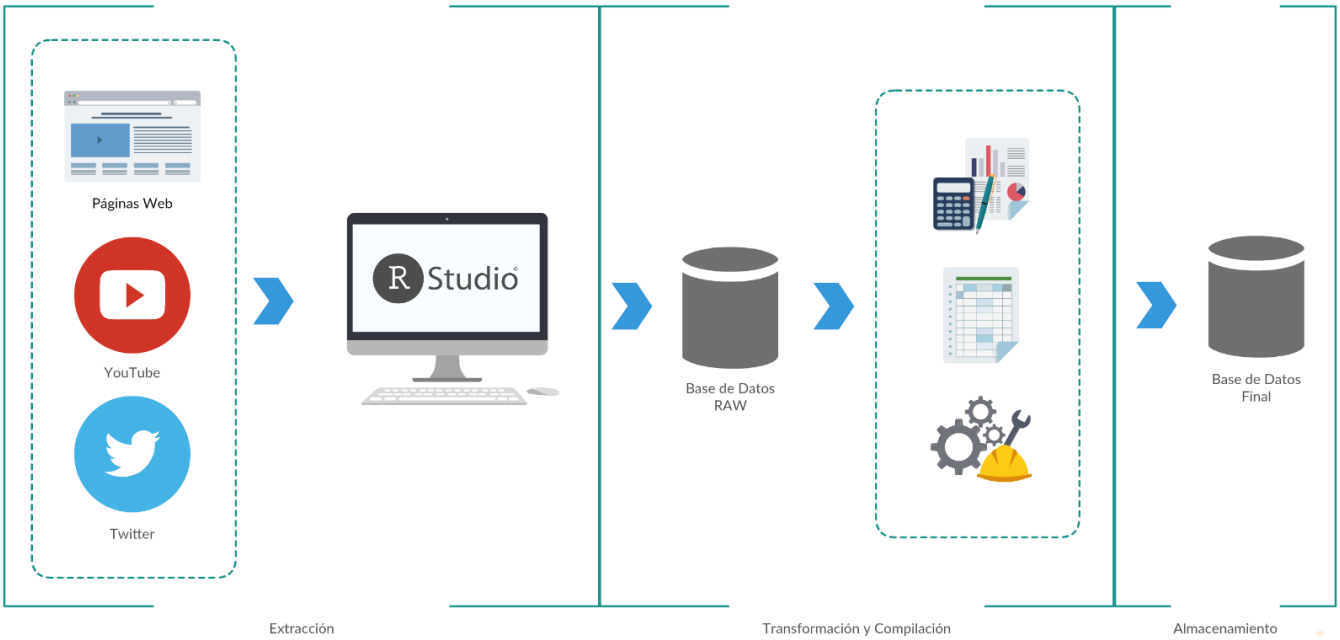
Es por eso que este trabajo va a desarrollar la extracción de informaciones diversas a través de canales diferentes; con el objetivo de crear un marco de actuación, previo a cualquier análisis de minería de datos.

1.2 ALCANCE DEL PROYECTO

El presente trabajo de fin de master comprenderá todos los aspectos necesarios para la creación de un compendio de datos del sector cinematográfico, utilizando información de libre disposición en la web. Este estudio se realizará sobre una muestra de películas estrenadas en Estados Unidos, en los años 2013, 2014 y 2015. (véase Anexo 1).

El mismo abarcará desde las tareas iniciales de selección y extracción de variables, tareas intermedias como la limpieza y estructuración de datos, y finalizará con el procedimiento de almacenamiento en una base de datos SQL (véase Imagen 1).

Imagen 1: Esquema General de Actividades.



1.3 OBJETIVOS

1.3.1 Objetivo General

- Selección, extracción y creación de un compendio de variables que permitan estudiar el comportamiento de la taquilla de películas en su semana de estreno, de acuerdo a variables de diferente índole, incluyendo variables de "ruido" social provenientes de redes sociales, mediante el uso y explotación de herramientas informáticas.

1.3.2 Objetivos Específicos

- Análisis de las variables extraídas directamente, y creación de variables alternativas, a partir de las directas.
- Limpieza de datos para mejorar la calidad de las variables.
- Almacenamiento de la base de datos obtenida, para su futura explotación.

2 SELECCIÓN DE LAS OBSERVACIONES Y EXTRACCIÓN DE DATOS

Para la creación de una base de datos significativa, se nos presentó la tarea de consolidar información de distintas fuentes, empleando diferentes técnicas de acuerdo a la disponibilidad de los datos. A continuación, detallaremos las distintas fuentes y las actividades realizadas en cada una de ellas:

2.1 BoxOfficeMojo

BoxOfficeMojo[1] es un sitio web que desde 1999 se dedica al seguimiento y conteo sistemático de los ingresos en taquilla que consiguen las películas, gracias a una serie de algoritmos. La web cubre los ingresos de taquilla semanales en más de cincuenta países.

Esta página fue utilizada para determinar el universo de estudio de este trabajo, al facilitar un listado de películas estrenada en los años cubiertos en el análisis (2013, 2014 y 2015).

La extracción de información de esta página web se realizó de manera directa, pues permite obtener todas las películas publicadas en un mismo año, de manera ordenada, y

[1] http://www.boxofficemojo.com/

con formato exportable a Excel. Así, se obtuvieron un total de 686 películas para 2013, 689 en 2014 y 495 en 2015 (ya que solo contamos con información hasta octubre). En total, el listado contenía 1.870 películas.

Tras un análisis básico manual previo, advertimos de que no todas las películas son válidas para el estudio, pues requiere preferentemente películas comerciales y que tengan alcance social. Sin embargo, muchas de las 1.870 del listado inicial son demasiado locales o alternativas, por lo que no existe información suficiente para rellenar la base de datos y poder incluirlas en los modelos.

Por ello, nos vimos obligados a realizar dos filtros previos a la extracción de los datos. En primer lugar, eliminamos aquellas películas estrenadas en pocas salas; esto ocurre bien porque la película no está concebida para grandes masas, o bien porque ciertos premios obligan a presentar la película en unas fechas concretas, y las productoras estrenan en esa fecha en unos pocos cines para poder competir en determinados festivales cinematográficos, pero no realizan la salida comercial hasta meses después (cuando sea más interesante para ella). En segundo lugar, se eliminaron las películas demasiado locales o triviales, y que por tanto no generaron suficiente "ruido" antes de su estreno en las redes sociales como para poder tenerlas en cuenta para el análisis. Tras este análisis previo, se estimó que el estudio definitivo se realizará sobre 425 películas.

Al determinar el universo de estudio, se procedió a la extracción de la información inicial, y por medio de un fichero CSV se obtuvieron las siguientes variables (véase tabla 1):

Tabla 1: Variables BoxOfficeMojo.

Variable	Descripción
NAME	Nombre completo con el que se ha comercializado la película
EST	Estudio que produjo la película
3D	Películas que principalmente fueron estrenadas en 3D
OC	Número de cines en los que se ha estrenado la película
Y	Ingresos de la película en el primer fin de semana de estreno

2.2 IMDB

Internet Movie Database[2] (IMDB; en español Base de Datos de Películas en Internet) es una base de datos en línea que almacena información relacionada con películas, personal del equipo de producción (incluyendo directores y productores), actores, series de televisión, programas de televisión, vídeojuegos, actores de doblaje y, más recientemente, personajes ficticios que aparecen en los medios de entretenimiento visual. Recibe más de 100 millones de usuarios distintos al mes y cuenta con una versión móvil. IMDB fue inaugurada el 17 de octubre de 1990, y en 1998 fue adquirida por Amazon.com.

A partir de la lista de películas obtenidas de BoxOfficeMojo, se procedió a recolectar la información necesaria disponible en IMDB. Esto se realizó mediante 3 técnicas: API (*Application Programming Interface*) de OMDB, *Web Scraping* directo y *Web Scraping* a través de *Web Archive*.

2.2.1 OMDBAPI

OMDBAPI[3] es un servicio web gratuito para obtener información sobre las películas. Dicha información proviene de IMDB, y es entregada en formato JSON mediante una solicitud web.

Para realizar esta solicitud, empleamos RStudio, y se requirió del paquete "rjson"[4] para poder interpretar la respuesta de la API (véase código 1).

Código 1: Librería utilizada en OMDBAPI.

```
install.packages("rjson")
library(rjson)
```

Para ello, se solicita la información utilizando la siguiente estructura (véase código 2)*:*

Código 2: Solicitud de información.

```
http://www.omdbapi.com/?i=ID&plot=short&r=json
```

[2] http://www.imdb.com/
[3] http://www.omdbapi.com/
[4] https://cran.r-project.org/web/packages/rjson/rjson.pdf

donde ID es el código de identificación, de acuerdo a IMDB, de la película solicitada. A esta consulta, el servicio web responde con un código JSON similar al siguiente (véase código 3):

Código 3: Ejemplo de información extraída de OMDBAPI para la película de Interstellar.

```
{"Title":"Interstellar","Year":"2014","Rated":"PG-
13","Released":"07 Nov 2014","Runtime":"169 min",
"Genre":"Adventure, Drama, Sci-Fi","Director":"Christopher
Nolan","Writer":"Jonathan Nolan, Christopher
Nolan","Actors":"Ellen Burstyn, Matthew McConaughey, Mackenzie
Foy, John Lithgow", "Plot":"A team of explorers travel through a
wormhole in space in an attempt to ensure humanity's
survival.","Language":"English","Country":"USA, UK","Awards":"Won
1 Oscar. Another 33 wins & 119
nominations.","Poster":"http://ia.media-
imdb.com/images/M/MV5BMjIxNTU4MzY4MF5BMl5BanBnXk
FtZTgwMzM4ODI3MjE@._V1_SX300.jpg","Metascore":"74","imdbRating":"
8.7","imdbVotes":"750,026","imdbID":"tt0816692","Type":"movie","R
esponse":"True"}
```

Empleando el paquete mencionado anteriormente, RStudio lo procesa como un objeto de tipo lista, el cual es fácilmente manipulable, para la obtención de la información deseada. A partir de ello, se extraen las siguientes variables (véase tabla 2):

Tabla 2: Variables extraídas de OMDBAPI

Variable	*Descripción*
MOVIE	Nombre de la película.
RATED	Clasificación de la película en función de la edad de recomendación.
RUNTIME	Duración de la película en minutos.
RELEASED	Fecha completa en la que se estrenó la película.
LANGUAGE	Idioma(s) en el(los) que se grabó la película. [MULTIVALUADO]
COUNTRY	País(es) en donde se grabó la película. [MULTIVALUADO]
GENRE	Género(s) en el(los) que se clasifica la película. [MULTIVALUADO]

Estas variables fueron almacenadas temporalmente en objetos de R para su posterior empleo.

2.2.2 Web Scraping

El *web scraping* es una metodología destinada a transformar datos procedentes de la web y sin estructura o semiestructurados en datos estructurados, los cuales pueden ser almacenados y analizados en una base de datos central, en una hoja de cálculo o en alguna otra fuente de almacenamiento. (Munzert, y otros, 2015)

Esta técnica se utilizó para obtener información complementaria, no disponible desde la API. Para esto utilizamos la aplicación RStudio y el paquete "rvest"[5] (véase código 4).

Código 4: Instalación de la librería 'rvest'.

```
install.packages("rvest")
library(rvest)
```

La metodología utilizada consistió en ubicar la ruta del código CSS, para cada una de las variables a extraer. Luego, utilizando RStudio, se extrajo el código HTML de la página donde se ubica la información de la película, y con ayuda del paquete "rvest" se procedió a tomar la información almacenada en cada una de las rutas CSS dentro de dicho HTML. De esta manera se obtuvieron las siguientes variables (véase tabla 3):

Tabla 3: Variables Web Scraping en IMDB

Variable	*Descripción*
DIR	Director de la película
S1	Nombre del 1er actor/actriz estrella de la película según IMDB
S2	Nombre del 2° actor/actriz estrella de la película según IMDB
S3	Nombre del 3er actor/actriz estrella de la película según IMDB
BUDGET	Presupuesto utilizado para la creación de la película
OPENINGWEEKEND	Ingresos de la película en el primer fin de semana de estreno

Estas variables fueron almacenadas temporalmente en objetos de R para su posterior empleo.

2.2.3 *Scraping* de *Web Archive*

Como ya se ha comentado anteriormente, el problema principal al que nos hemos enfrentado y que ha generado ciertas limitaciones en el alcance del trabajo, es la

[5] https://cran.r-project.org/web/packages/rvest/rvest.pdf

importancia de la temporalidad de ciertas variables. Esto se refiere a que estas variables deben ser medidas (capturadas) en un periodo anterior al estreno de la película, lo cual constituye un problema, ya que las fuentes no suelen almacenar datos históricos, sino datos actualizados.

Para solucionar este contratiempo, recurrimos a *Web Archive*. *Web Archive*[6] está impulsado por una empresa sin ánimo de lucro cuyo objetivo al respecto es crear una librería digital que provea acceso público a una colección de material histórico digitalizado, incluyendo páginas webs, softwares, juegos, música o libros. En 2014 alcanzaron los 15 Petabytes de información, con un crecimiento mensual de unos 20 Terabytes. Una parte de esta organización se denomina *The Wayback Machine*. Este archivo web contiene alrededor de 150 billones de capturas webs desde su creación en 1996. Y fue esta web, con ayuda de su API, la que nos permitió obtener la información requerida, con el factor temporal que necesitamos.

La API de disponibilidad de *Web Archive*, partiendo de una URL y una fecha de solicitud, devuelve el URL donde se encuentra almacenada la captura de la página para la fecha (o en caso de no existir, retorna el URL asociado a la fecha más cercana). Para utilizar esta API hay que enviar las solicitudes al siguiente enlace (véase código 5):

Código 5: Conexión con la API de Web Archive.

```
http://archive.org/wayback/available?url=example.com&timestamp=YYYYMMDD
```

Donde example.com corresponde al URL que se desea solicitar, y YYYYMMDD a la fecha en que se desea visualizar la captura del URL en formato año (4 dígitos), mes (2 dígitos) y día (2 dígitos). Tras esta solicitud, el servicio web responderá con un código JSON con la siguiente estructura (véase código 6):

Código 6: Respuesta JSON API disponibilidad en Web Archive

```
{
    "archived_snapshots": {
        "closest": {
            "available": true,
```

[6] https://archive.org/about/

```
            "url":
"http://web.archive.org/web/20060101064348/http://www.exampl
e.com:80/",
            "timestamp": "20060101064348",
            "status": "200"
        }
    }
}
```


Luego, procedimos a leer esta respuesta utilizando RStudio, y el paquete "JSON" mencionado anteriormente, y de esta manera obtuvimos la URL de donde extrajimos el código HTML de la página capturada, y de manera similar al proceso anterior, aplicamos *scraping* sobre la captura para obtener las siguientes variables con componentes temporales (véase tabla 4):

Tabla 4: Variables de IMDB extraídas de Web Archive

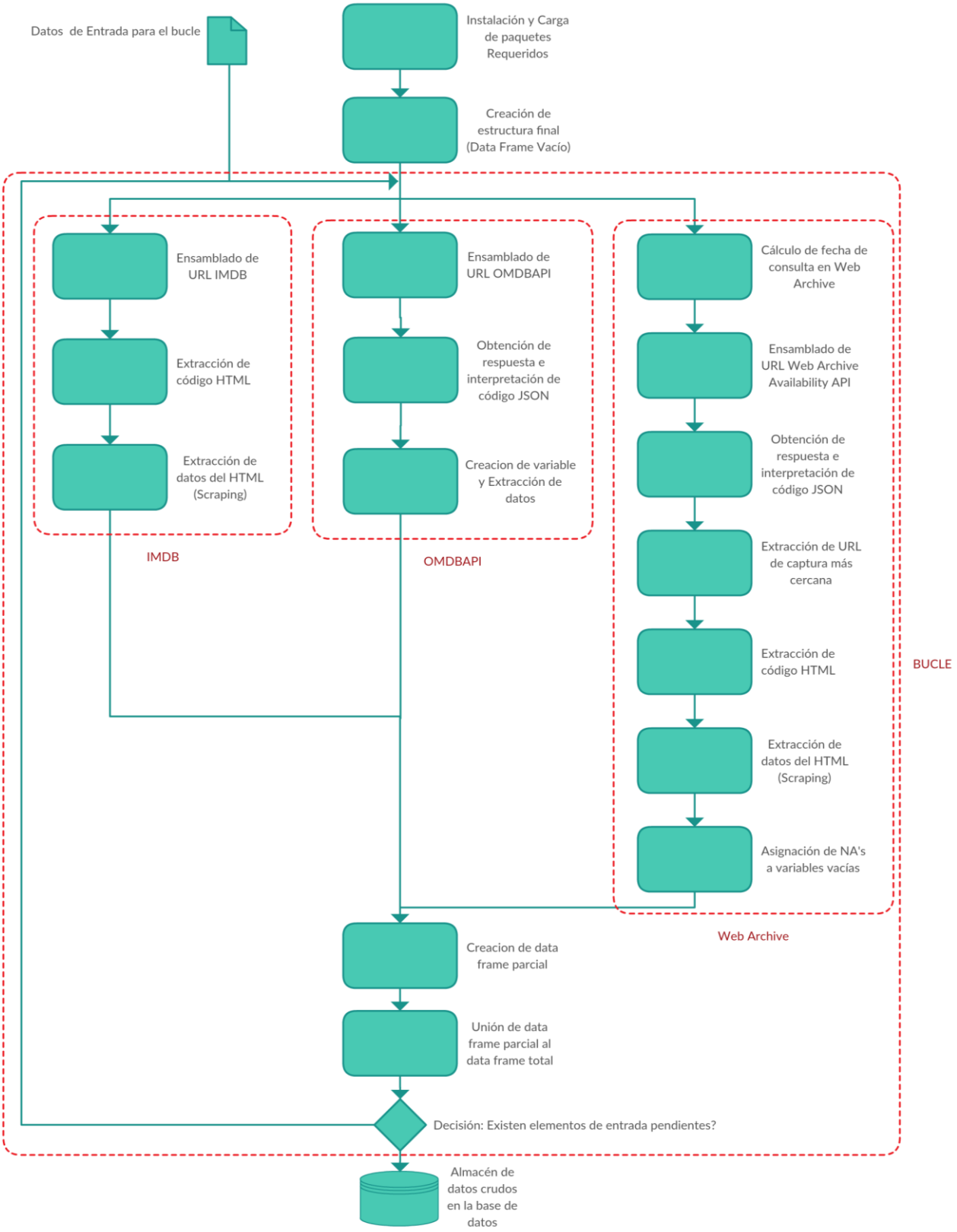
Variable	*Descripción*
TS10DB	*Timestamp* de la captura encontrada
IMDBRT10DB	Puntuación de la película según los usuarios de IMDB
IMDBUSR10DB	Cantidad de usuarios que han puntuado la película
MS10DB	Puntuación de la película según los críticos de IMDB
RVUSR10DB	Cantidad de usuarios que han realizado *reviews* de la película
RVCRT10DB	Cantidad de críticos que han realizado *reviews* de la película
MTCRT10DB	Cantidad de críticos que han puntuado la película
MOVIEMETER10DB	Indicador de popularidad de la película según IMDB

Estas variables fueron almacenadas temporalmente en objetos de R para su posterior procesamiento.

2.2.4 Consolidación de Archivos

Luego de estas 3 tareas se obtuvieron una serie de variables almacenadas temporalmente en objetos de R, y procedimos a crear un registro para cada película mediante la unión de estos objetos. De esta manera se obtuvo un *dataframe* con la información de todas las películas (véase imagen 2).

Imagen 2: Diagrama de Proceso: Extracción de datos de IMDB.


Datos de Entrada para el bucle
Instalación y Carga de paquetes Requeridos
Creación de estructura final (Data Frame Vacío)
Ensamblado de URL IMDB
Extracción de código HTML
Extracción de datos del HTML (Scraping)
IMDB
Ensamblado de URL OMDBAPI
Obtención de respuesta e interpretación de código JSON
Creacion de variable y Extracción de datos
OMDBAPI
Cálculo de fecha de consulta en Web Archive
Ensamblado de URL Web Archive Availability API
Obtención de respuesta e interpretación de código JSON
Extracción de URL de captura más cercana
Extracción de código HTML
Extracción de datos del HTML (Scraping)
Asignación de NA's a variables vacías
Web Archive
BUCLE
Creacion de data frame parcial
Unión de data frame parcial al data frame total
Decisión: Existen elementos de entrada pendientes?
Almacén de datos crudos en la base de datos

2.3 Twitter

Twitter[7] es un servicio de *microblogging*, creado en California en marzo de 2006. La red es mundialmente conocida: se estima que tiene más de 500 millones de usuarios, 65 millones de *tweets* al día y más de 800 000 peticiones de búsqueda diarias. La red permite enviar mensajes de texto plano de corta longitud, con un máximo de 140 caracteres, llamados *tweets*, que se muestran en la página principal del usuario. Los usuarios pueden suscribirse a los *tweets* de otros usuarios, que pueden ser marcados como favoritos y/o redifundidos (*retweet*). Por defecto, los mensajes son públicos, pudiendo difundirse de una forma más restringida y mostrarse únicamente a unos seguidores determinados.

Para utilizar esta red social como fuente, establecimos 2 enfoques: (1) El análisis de la cuenta oficial para mostrar lo que la productora quiere ofrecer a los usuarios; y (2) La búsqueda por el *hashtag* que los usuarios utilizan para mostrar su opinión a otros usuarios.

2.3.1 Extracción y explotación de *tweets* de cuentas oficiales

En esta etapa utilizamos RStudio y el paquete "twitteR"[8] como paquete principal, y los paquetes "RCurl", "RJSONIO" y "stringr" como paquetes auxiliares (véase código 7).

Código 7: Librerías necesarias para conectar con Twitter.

```
install.packages("twitteR", "RCurl", "RJSONIO", "stringr")
library(twitteR)
library(RCurl)
library(RJSONIO)
library(stringr)
```

Cabe recalcar que el paquete "twitteR" permite descargar el *timeline* (línea de tiempo: manera de mostrar una serie de eventos (*tweets*) en orden cronológico) de un usuario hasta un máximo de 3200 *tweets* (incluyendo *retweets* hechos por el mismo) y no permite establecer un límite temporal a la solicitud de *tweets*.

[7] https://about.twitter.com/es/company
[8] https://cran.r-project.org/web/packages/twitteR/twitteR.pdf

Dicho esto, procedimos a ubicar las cuentas oficiales de *Twitter* de cada una de las películas, para luego realizar la descarga de su *timeline* y encontramos 3 casos:

1. Cuentas oficiales de la película

En este grupo se encuentran la gran mayoría de las películas, pues para casi todas ellas se creó una cuenta exclusiva en la mencionada red social para la promoción de la película. Este caso no presentó mayor inconveniente, ya que las cuentas, por su propia naturaleza de haber sido creadas para la promoción de una película, no alcanzaban el límite establecido por el paquete "twitteR", y se pudo obtener la totalidad de los *tweets* perteneciente a cada una de las cuentas.

Por otra parte, como ya se mencionó anteriormente, el paquete no permite establecer un intervalo temporal para la descarga de *tweets*, por lo que tras su descarga se debió aplicar un paso adicional para excluir aquellos que fueron realizados en los 10 días previos al estreno de la película.

2. Cuentas oficiales de la productora

En un segundo grupo encontramos aquellas películas que fueron promocionadas directamente desde la cuenta oficial del estudio que produjo la película. Esto representó un problema para la extracción de datos, ya que estas cuentas siempre superaban el límite de 3200 *tweets*, por lo cual las consultas no devolvían *tweets* lo suficientemente antiguos para este estudio. Adicionalmente, estas cuentas se encargaban de promover varias películas simultáneamente, lo que nos impidió extraer métricas particulares para cada una de las películas de este grupo.

3. Películas sin presencia en Twitter

En esta etapa del estudio nos encontramos con un último caso, en donde tenemos a un reducido grupo de películas que no tienen presencia en Twitter.

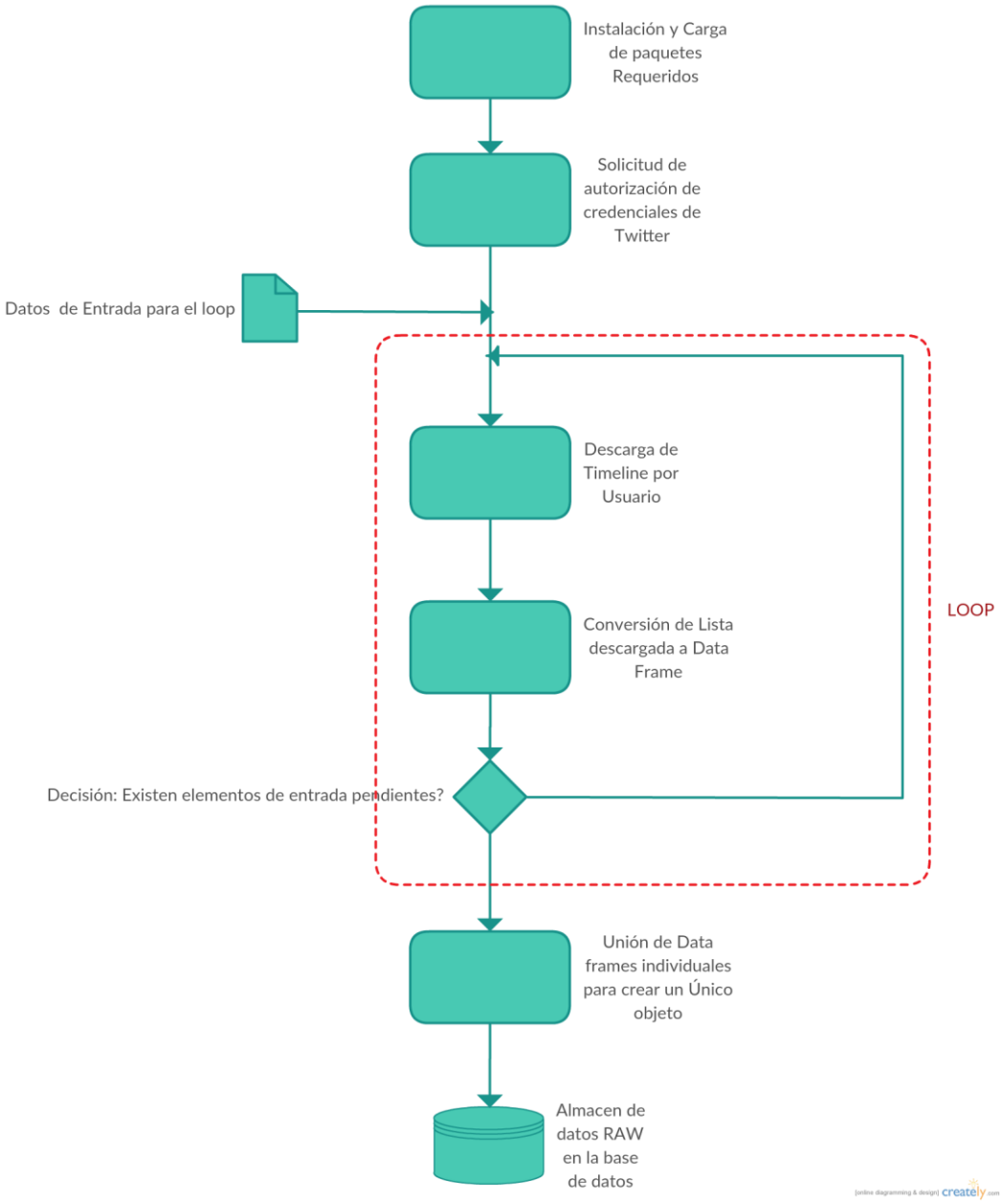
Teniendo en cuenta este escenario, se extrajeron únicamente los *tweets* de las cuentas del primer grupo. Del resto, considerando que no existe una cuenta destinada a la promoción de la película, se incluyeron también con las variables informadas, pero inicializadas con el valor cero (0). De aquí se obtuvieron las siguientes variables (véase tabla 5):

Tabla 5: Variables obtenidas de las cuentas oficiales de Twitter para cada *tweet*

Variable	*Descripción*
TEXT	Texto del *tweet* extraído
FAVORITECOUNT	Número de veces que el *tweet* ha sido marcado como favorito
CREATED	Fecha de creación del *tweet*
SCREENNAME	Nombre del usuario autor del *tweet*
RETWEETCOUNT	Número de veces que el *tweet* ha sido retuiteado
ISRETWEET	Indicador binario de si el *tweet* es original del autor o es un *retweet* de otro autor

Estas variables se fueron almacenando en objetos temporales de R, para luego crear un registro para cada una de las películas, y mediante un bucle se creó un *dataframe* con la información de la totalidad de las películas (véase imagen 3).

Imagen 3: Diagrama de Proceso: Extracción de datos de las cuentas oficiales promotoras de las películas en Twitter.



2.3.2 Extracción y explotación de *tweets* de usuarios

La etapa adicional de extracción de información de Twitter se centró en la búsqueda de *tweets* utilizando un término de búsqueda o etiqueta (conocido en la red social como *hashtag*). En esta etapa encontramos otra limitación, ya que el paquete "twitteR" solo permite la extracción de *tweets* individuales en un periodo no superior a 7 días, y para nuestro estudio necesitábamos, dependiendo de la película, *tweets* de incluso 2 años de antigüedad.

Esto lo solventamos mediante el uso de los servicios de Topsy[9], empresa que se encarga de almacenar y comercializar la totalidad de *tweets* generados en la red social.

La empresa nos facilitó acceso gratuito a sus archivos, en un periodo de prueba de 30 días, mediante su API[10], con la limitación de solo poder recuperar un máximo de 9000 *tweets* diarios, y hasta 500 *tweets* en cada consulta. Teniendo en cuenta este límite y la cantidad de películas incluidas en nuestro estudio, decidimos extraer una cantidad máxima de 401 *tweets* por película.

Las solicitudes a su API se realizan mediante la siguiente URL (véase código 8):

Código 8: Conexión con la API de Topsy

```
http://api.topsy.com/v2/content/tweets.format?query=parameters
```

Donde los parámetros pueden adoptar los siguientes valores (véase tabla 6):

Tabla 6: Opciones de información disponibles en Topsy.

Nombre	Requerido	Descripción
q	*Opcional*	Devuelve los resultados que concuerdan con el patrón de búsqueda. Si se omite, se muestran todos los *tweets*. Si se buscan varios patrones, deben delimitarse con comas (siendo la coma 'url-encoded').
sort_by	*Opcional*	Método de ordenación de la lista de resultados
offset	*Opcional*	Entero que indica la distancia (desplazamiento) desde el inicio del objeto.
limit	*Opcional*	Número máximo de resultados devuelto.
include_metrics	*Opcional*	Para cada resultado incluye una serie temporal con citas, así como métricas de *trending*, aceleración, y picos, cuando include_metrics=1; por defecto es 0. Estas métricas tienen un coste adicional.
include_enrichment_all	*Opcional*	Cuando es igual a 1, para cada resultado, Topsy enriquece los campos (URLs completas, geolocalización, sentimiento e influencia) (por defecto es igual a 0). Estas métricas tienen un coste adicional.
new_only	*Opcional*	Solo incluye resultados desde un primer *tweet* en un *timeframe* especificado cuando =1; por defecto es 0.
mintime	*Condicion*	Fecha al comienzo de la extracción en formato Unix *timestamp*. Se puede

[9] http://topsy.com/
[10] http://api.topsy.com/doc/

	al	combinar con *slice* y *maxtime*. Obligatorio si *maxtime* es solicitado.
maxtime	*Opcional*	Fecha final de la extracción en formato Unix *timestamp*. Por defecto es el momento en el que la extracción se ha solicitado.
region	*Opcional*	Muestra el resultado generados desde una localización concreta. Debe usarse el ID de cada región, que puede encontrarse usando el recurso /locationl. Puede incluirse una lista de regiones a través de comas.
latlong	*Opcional*	Muestra el resultado de los *tweets* geolocalizados con longitud/latitud. Disponible cuando "latlong=1"; por defecto es 0.
allow_lang	*Opcional*	Muestra resultados expresados en un idioma concreto. Las opciones son: 'en' (English), 'zh' (Chino), 'ja' (Japonés), 'ko' (Coreano), 'ru' (Ruso), 'es' (Español), 'fr' (Francés), 'de' (Alemán), 'pt' (Portugués), y 'tr' (Turco).
sentiment	*Opcional*	Muestra resultados con un sentimiento específico. Las opciones son: 'pos' (Positivo), 'neu' (Neutral) o 'neg' (Negativo).
infonly	*Opcional*	Muestra únicamente *tweets* de usuarios influyentes. Disponible con 'infonly=1'; por defecto es 0.
tweet_types	*Opcional*	Muestra algún tipo de *tweet* concreto. Las opciones son: '*tweet*', '*reply*', '*retweet*'. Por defecto se incluyen todos; y no puede aplicarse al recurso *insights/influencers*.

La respuesta es entregada en formato JSON, la cual. Con ayuda del paquete "RJSON", la convertimos en un objeto de tipo lista en RStudio. A partir de esta respuesta, creamos un *dataframe* temporal, que servirá para estructurar la información. Este *dataframe* estará compuesto por las siguientes variables (véase tabla 7):

Tabla 7: Variables obtenidas de *hashtags* y términos de búsqueda en Twitter

Variable	*Descripción*
ID	Incluye tanto el *hashtag* de la película como términos de búsquedas utilizados por los *community managers*.
ID.IMDB	ID de IMDB que sirve como ID general de la base de datos para la película
USER	Nombre del usuario que ha escrito el *tweet*

INFLUENCE	Influencia del usuario que ha escrito el *tweet*, presentado en una escala del 1 al 10
ID.*TWEET*	ID interno que maneja Twitter para controlar cada texto escrito
TWEET	Texto propiamente dicho que se ha escrito, enviado y no se ha borrado
URL.*TWEET*	Link web con el que acceder al *tweet* desde un buscador.
CITATIONS	Sumatorio de *retweets*, citas y respuestas a un *tweet*.
IMPRESSIONS	Número de veces que un *tweet* ha sido visto por algún usuario.

- Análisis de Sentimientos

Como complemento a esta etapa nos propusimos a realizar un análisis de sentimientos sobre los *tweets* extraídos en el paso anterior. Esto lo realizamos mediante un estudio de polaridad para cada *tweet*. Para realizar la asignación de dicha polaridad, requerimos de diccionarios especializados que nos posibiliten la tarea de contrastar nuestro texto con una referencia léxica para el idioma inglés[11].

Para esto utilizaremos los paquetes (véase código 9):

Código 9: Librerías necesarias para el análisis de sentimientos

```
install.packages("stringr", "plyr")
library(stringr)
library(plyr)
```

Previamente preparamos el texto del *tweet* para el análisis de sentimientos, utilizando el siguiente código en RStudio:

Código 10: Limpieza de los textos de twitter.

```
# remover entidades de retweet, personas, puntuación, control,
números, enlaces HTML y espacios innecesarios:
txt = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", txt)
txt = gsub("@\\w+", "", txt)
txt = gsub("[[:punct:]]", "", txt)
txt = gsub('[[:cntrl:]]', '', txt)
txt = gsub("[[:digit:]]", "", txt)
txt = gsub("http\\w+", "", txt)
```

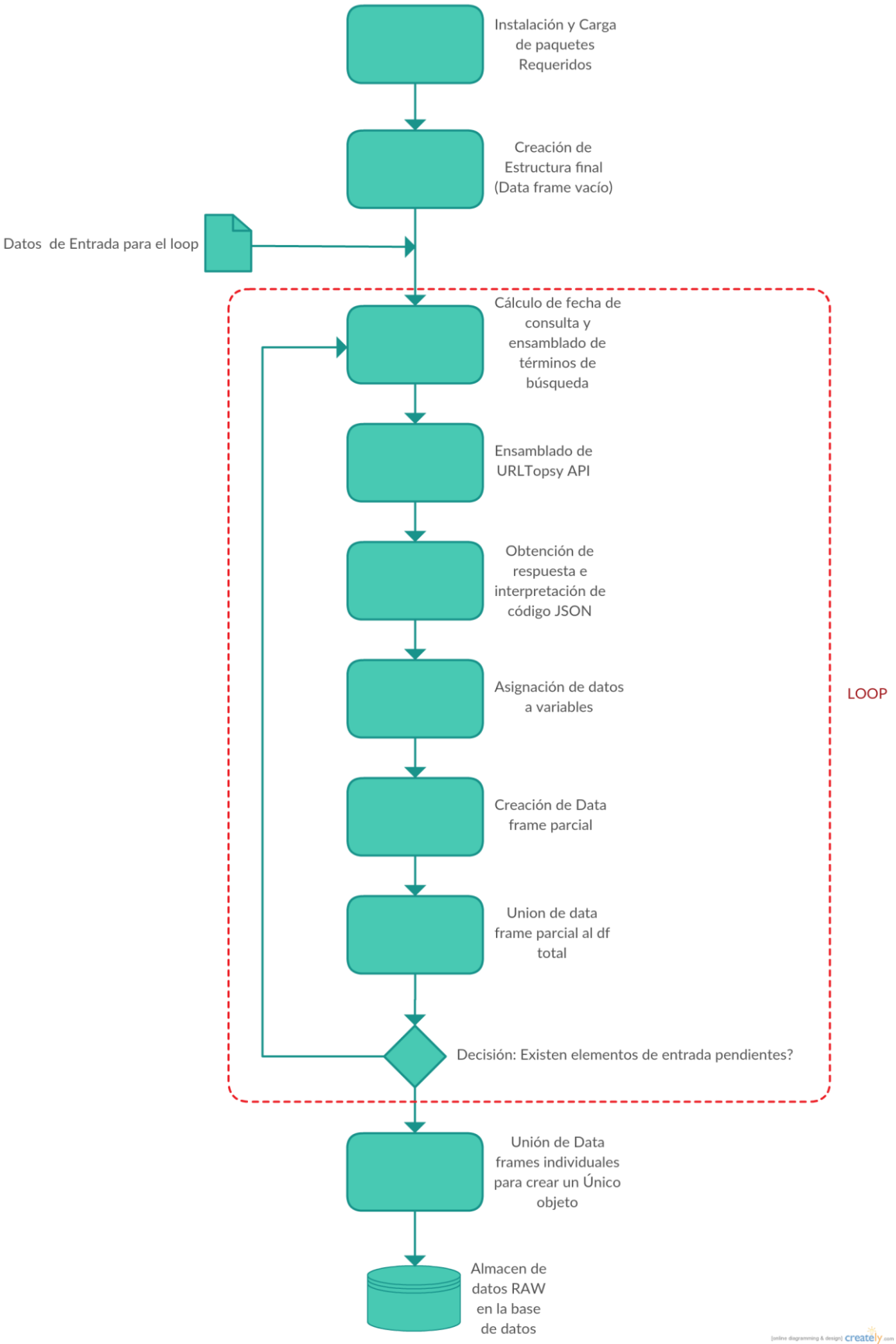
[11] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

```
txt = gsub("[ \t]{2,}", "", txt)
txt = gsub("^\\s+|\\s+$", "", txt)

# convertimos todo el texto a minúscula:
txt =  tolower(txt)

# Con el paquete string, separamos las palabras:
word.list  =  str_split(txt, '\\s+')
```

Luego de esto, con los diccionarios previamente cargados, procedemos a evaluar la lista de palabras de cada *tweet*. De esta manera obtuvimos una puntuación de polaridad para cada *tweet*, la cual clasifica el texto como: positivo, neutro o negativo.

Este valor de polaridad fue añadido como una columna nueva al *dataframe* original, caracterizando cada uno de los *tweets*.

Imagen 4: Diagrama de Proceso: Extracción Twitter Cuentas Usuarios.


Instalación y Carga de paquetes Requeridos
Creación de Estructura final (Data frame vacío)
Datos de Entrada para el loop
Cálculo de fecha de consulta y ensamblado de términos de búsqueda
Ensamblado de URLTopsy API
Obtención de respuesta e interpretación de código JSON
Asignación de datos a variables
Creación de Data frame parcial
Union de data frame parcial al df total
Decisión: Existen elementos de entrada pendientes?
LOOP
Unión de Data frames individuales para crear un Único objeto
Almacen de datos RAW en la base de datos

2.4 YouTube

Como última fuente, utilizamos la plataforma de vídeos en línea YouTube. Para obtener los datos de YouTube, nuevamente nos encontramos con la limitación temporal de nuestro estudio. Es por ello que decidimos acudir de nuevo a Web Archive como fuente de capturas de YouTube, para aplicar nuevamente la técnica del *scraping* sobre la misma.

En este caso tuvimos que contar además con otro problema añadido: las variaciones que YouTube ha realizado a su página web durante los últimos 3 años. requirió reajustar constantemente las rutas CSS para acceder a la información, y poder realizar la extracción de datos (*scraping)*.

Similar al proceso utilizado anteriormente para *scraping*, el paso inicial es identificar aquellas variables que serán parte de la extracción y su ruta CSS, para poder obtenerlas mediante el código HTML de la página. Seguidamente, tuvimos que recopilar los identificadores de los vídeos que formaron parte del análisis (correspondiente a cada una de las películas incluidas en el mismo). Para esto realizamos la búsqueda del tráiler oficial de cada película, y ordenamos los resultados por mayor número de visitas. El resultado con mayor cantidad de visitas fue el seleccionado para la extracción de información.

Luego, con ayuda del API de disponibilidad de *Web Archive*, consultamos la disponibilidad de la captura más cercana a la del URL del vídeo, y utilizando el URL de respuesta, aplicamos *scraping* sobre la misma.

Es importante destacar que la página web de YouTube utiliza Javascript en su código, por lo cual, para poder leer este código fue necesario un intérprete. Para realizar esta tarea utilizamos el paquete "RSelenium"[12] (véase código 11), fue necesario, en conjunción con la utilidad "PhantomJS"[13], como aplicación emuladora del navegador e intérprete de Javascript.

Código 11: Librerías utilizadas para extraer la información de YouTube.

[12] https://github.com/ropensci/RSelenium
[13] http://phantomjs.org/

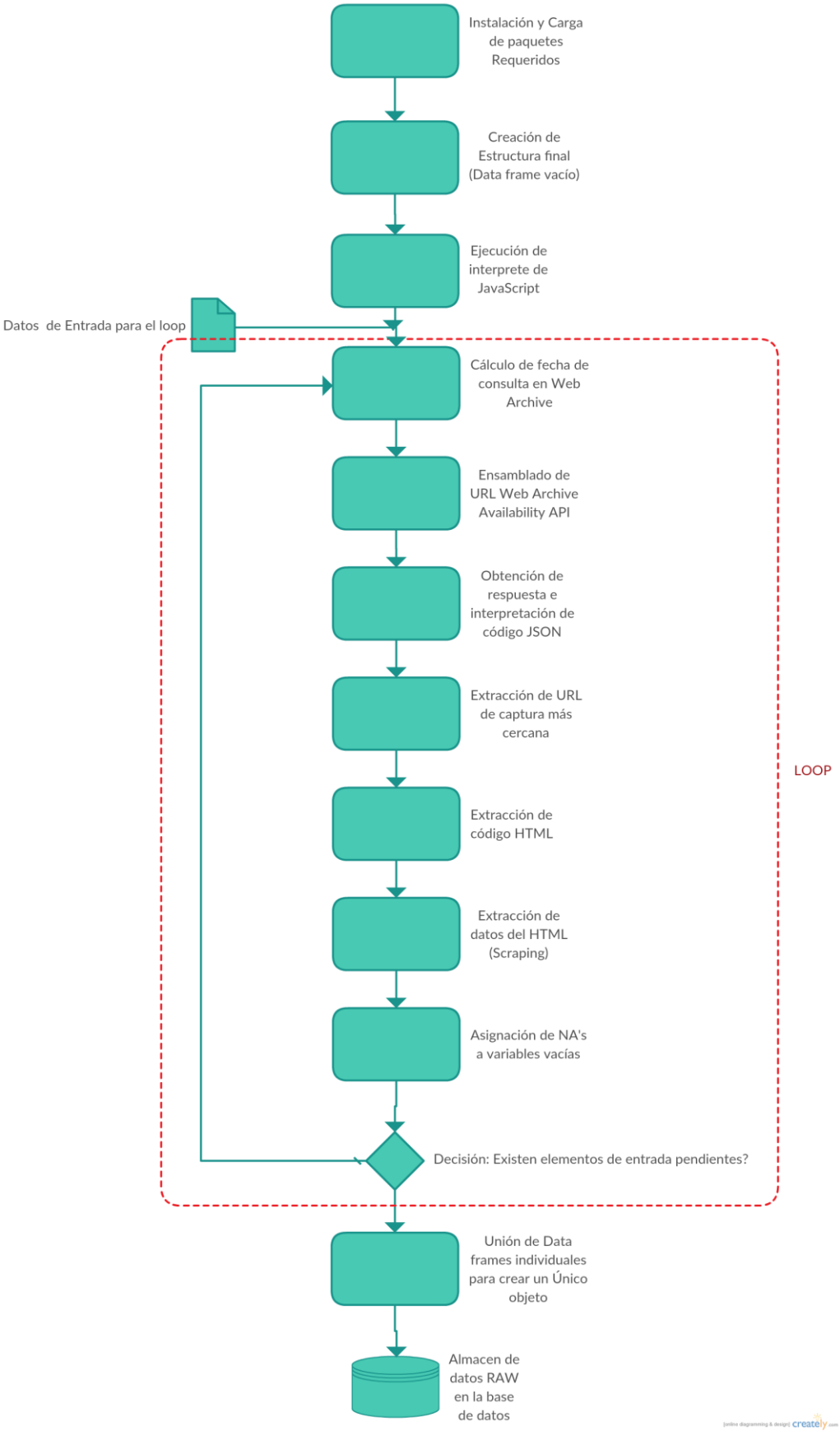
```
install.packages("devtools")
devtools::install_github("ropensci/RSelenium")
library(RSelenium)
```

Este proceso se completa, al utilizar un bucle para recolectar la información de cada película, para luego anexarla a un *dataframe*. Al final de este proceso obtuvimos las siguientes variables (véase tabla 8):

Tabla 8: Variables extraídas de YouTube.

Variable	*Descripción*
ID.IMDB	ID de control para el registro, extraído de IMDB
MOVIE	Nombre de la película
ID.YOUTUBE	ID del vídeo en YouTube
CANAL.YT	Canal que subió el vídeo a YouTube
FECHA.PUB	Fecha de publicación del vídeo
FSNAP.10DB	Fecha en la que se realizó la captura de la página web
SUBSCANAL.10DB	Cantidad de suscriptores del canal
VIEWS.10DB	Número de visualizaciones del vídeo
LIKES.10DB	Número de personas a las que les gustó el vídeo
DISLIKES.10DB	Número de personas a las que no les gustó el vídeo

Imagen 5: Diagrama de Proceso: Extracción YouTube.


Instalación y Carga de paquetes Requeridos
Creación de Estructura final (Data frame vacío)
Ejecución de interprete de JavaScript
Datos de Entrada para el loop
Cálculo de fecha de consulta en Web Archive
Ensamblado de URL Web Archive Availability API
Obtención de respuesta e interpretación de código JSON
Extracción de URL de captura más cercana
Extracción de código HTML
Extracción de datos del HTML (Scraping)
Asignación de NA's a variables vacías
Decisión: Existen elementos de entrada pendientes?
LOOP
Unión de Data frames individuales para crear un Único objeto
Almacen de datos RAW en la base de datos

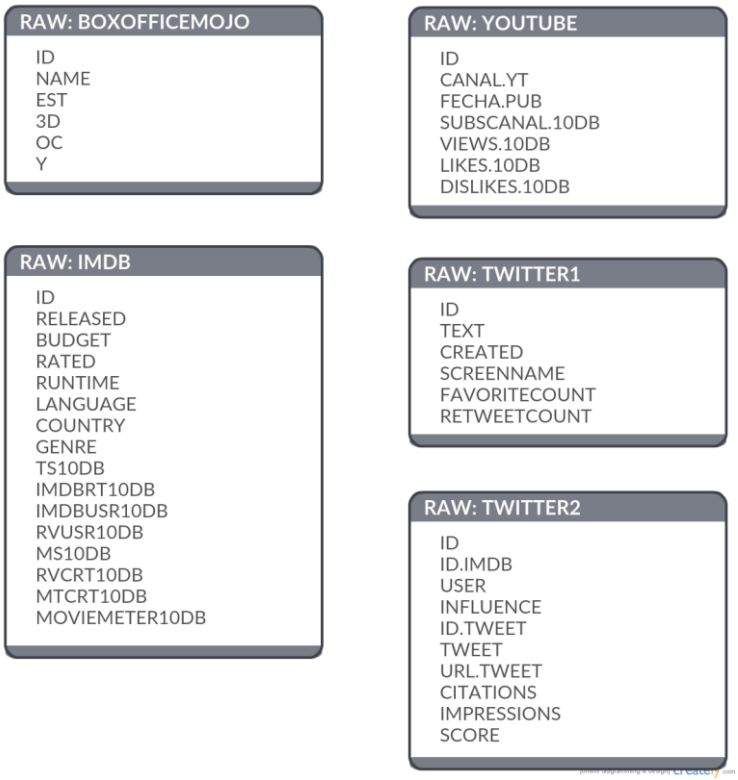
3 PREPARACIÓN DE DATOS CRUDOS: Limpieza y estructuración de datos: Feature extraction and Feature Engineering

Decidir cuáles son las variables que deben utilizarse para la realización de análisis también puede involucrar una fase de experimentación previa. Esta fase del proceso se conoce como Extracción de Características (Feature Extraction) e/o Ingeniería de Características (Feature Engineering).

De acuerdo con Bowles (2015), la extracción de características es el proceso de tomar datos en un arreglo de forma libre (o sin formato aparente) y estructurarlos. La ingeniería de características es el proceso de manipulación y combinación de características para llegar a otras más informativas.

Luego de haber generado las tablas de información, procedimos a limpiar y estructurar los datos crudos (*raw*). Este proceso permitió que la información recabada pudiera ser fácilmente utilizada para la generación de conocimientos. Para la ejecución de esta labor, se aplicaron actividades como la codificación de variables nominales con códigos numéricos, el renombramiento de variables o la creación de nuevas variables a partir de las variables originales, entre otras.

Imagen 6: Esquema de datos crudos.



Estas tareas fueron aplicadas a cada conjunto de datos de la siguiente manera:

a. BoxOfficeMojo

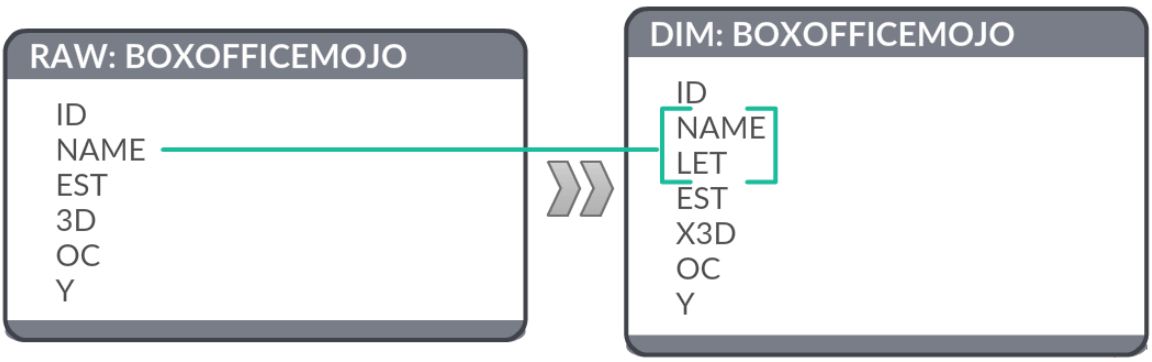
A esta fuente de datos le fueron aplicadas las actividades de renombramiento de variables y de codificación de variables nominales. Adicionalmente, se creó una nueva variable. El resultado fue el siguiente (véase tabla 9 e imagen 7):

Tabla 9: Preparación de Variables BoxOfficeMojo.

Variable Origen	*Variable Nueva*	*Descripción*
NAME	***NAME***	Nombre completo con el que se ha comercializado la película
	LET	Número de letras que componen el nombre
EST	***EST***	Estudio que produce la película Esta variable fue recodificada con valores numéricos.
3D	***X3D***	Películas que fueron estrenadas principalmente en 3D Esta variable fue recodificada con valores binarios.
OC	***OC***	Número de cines en los que se ha estrenado una película
Y	***Y***	Ingresos de una película en el primer fin de semana de estreno

No se modificó | Variable generada a partir de la original | Variable eliminada

Imagen 7: Esquema de base de datos de BoxOfficeMojo.



b. IMDB

Las variables de esta fuente fueron las que sufrieron más modificaciones en su estructura. En efecto hubo que crear numerosas variables nuevas para sintetizar la información contenida en ellas. A continuación, se muestra un resumen de las actividades aplicadas sobre la misma (véase tablas 10, 11 y 12 e imagen 8):

- **OMDBAPI**

Tabla 10: Preparación de Variables OMDBAPI.

Variable Origen	*Variable Nueva*	*Descripción*
MOVIE	***NAME***	Nombre de la película
RATED	***RAT***	Clasificación de la película en función de la edad de recomendación. Esta variable fue recodificada con valores numéricos.
RUNTIME	***RUNT***	Duración de la película en minutos
RELEASED	***DATA***	Fecha completa en la que se estrenó la película. Esta variable fue descompuesta en 3 variables, y eliminada.
	YEAR	Año en que se estrenó la película

		MON	Mes en que se estrenó la película. Esta variable fue codificada con valores enteros (1:12)
		DAY	Día del mes en que se estrenó la película
LANGUAGE		***LANGUAGE***	Idioma(s) en el(los) que se grabó la película. A partir de esta variable se crearon 3 nuevas, y la original fue eliminada.
		L_N	Número de idiomas en los que se desarrolla la película
		L_E	Indicador binario: incluye inglés
		L_O	Indicador binario: incluye algún otro idioma principal (ruso, alemán, francés o japonés). Estos idiomas principales se determinaron, a partir de la frecuencia de los mismos en la muestra, donde se seleccionaron aquellos que tuvieran una presencia significativa en la misma.
COUNTRY		***COUNTRY***	País(es) en donde se grabó la película A partir de esta variable se crearon 5 nuevas variables, y la original fue eliminada.
		C_N	Número de países donde se grabó la película.
		C_USA	Indicador binario: la locación de la película está restringida exclusivamente a EEUU
		C_EU	Indicador binario: la locación de la película incluye Europa
		C_RA	Indicador binario: la locación de la película incluye América (sin EEUU)
		C_AO	Indicador binario: la locación de la película incluye Asia u Oceanía

GENRE		***GENRE***	Género(s) en el(los) que se clasifica la película A partir de esta variable se crearon 11 nuevas variables, y la original fue eliminada.
		G_N	Número de géneros en los que se clasifica la película
		G_D	Indicador binario: Género Drama
		G_C	Indicador binario: Género Comedia
		G_AC	Indicador binario: Género Acción
		G_T	Indicador binario: Género Thriller
		G_F	Indicador binario: Género Fantasía o SciFi
		G_R	Indicador binario: Género Romance
		G_B	Indicador binario: Género Biografía
		G_AN	Indicador binario: Género Animación
		G_H	Indicador binario: Género Horror
		G_M	Indicador binario: Género Musical

No se modificó Variable generada a partir de la original Variable eliminada

- **Scraping IMDB**

Tabla 11: Preparación de Variables Scraping IMDB

Variable Origen	*Variable Nueva*	*Descripción*
DIR	***DIR***	Variables desestimadas Estas variables fueron desestimadas, ya que la intención era crear una valoración de los artistas, mediante la cuantificación de su impacto al momento del estreno de la película. Pero todos los planteamientos sugeridos fueron imposibilitados por la falta de información histórica que nos permitiera realizar la mencionada evaluación.
S1	***S1***	
S2	***S2***	
S3	***S3***	
BUDGET	***BUDG***	Presupuesto utilizado para la creación de la película
OPENINGWEEKEND	***Y***	Ingresos de una película en el primer fin de semana de estreno

No se modificó | Variable generada a partir de la original | Variable eliminada

- **Scraping Web Archive**

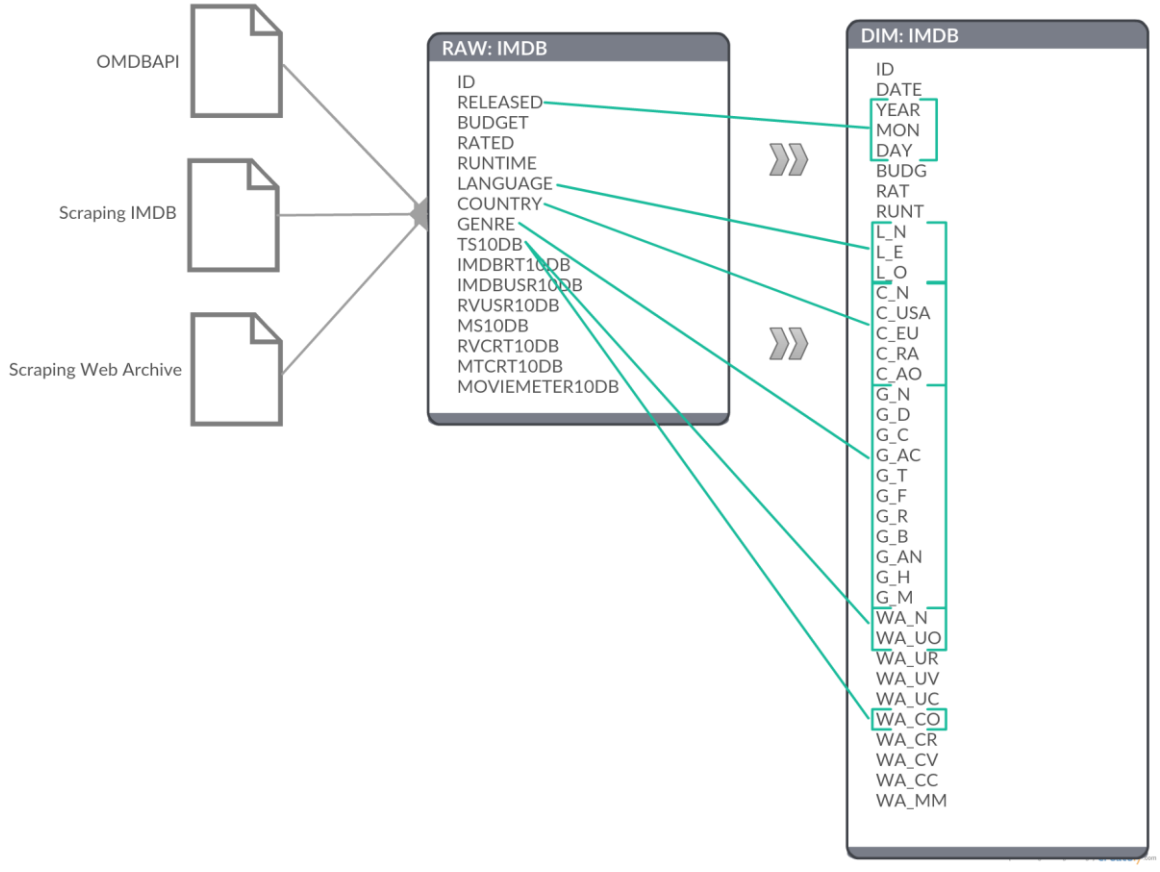
Tabla 12: Preparación de las variables procedentes del *scraping* de Web Archive

Variable Origen	*Variable Nueva*	*Descripción*
TS10DB	***TS10DB***	Fecha de la captura web almacenada en *Web Archive* Variable eliminada.
	WA_N	Número de capturas encontradas en *Web Archive* en 4 fechas distintas anteriores al estreno
IMDBRT10DB	***WA_UO***	Indicador binario: Factibilidad de votación para usuarios hasta 10 días antes del estreno
	WA_UR	Valoración de los usuarios de IMDB hasta 10 días antes del estreno
IMDBUSR10DB	***WA_UV***	Número de usuarios que han votado en IMDB hasta 10 días antes del estreno

RVUSR10DB	***WA_UC***	Número de críticas que han realizado los usuarios hasta 10 días antes del estreno
MS10DB	***WA_CO***	Indicador binario: Factibilidad de votación para críticos hasta 10 días antes del estreno
	WA_CR	Valoración de los críticos hasta 10 días antes del estreno
RVCRT10DB	***WA_CC***	Número de *reviews* que han realizado los críticos hasta 10 días antes del estreno
MTCRT10DB	***WA_CV***	Número de críticos que han votado en IMDB hasta 10 días antes del estreno
MOVIEMETER10DB	***WA_MM***	Popularidad de la película hasta 10 días antes del estreno Esta variable se categorizó para homogeneizar sus características

No se modificó | Variable generada a partir de la original | Variable eliminada

Imagen 8: Representación de la base de datos con información de IMDB



c. Twitter

Esta etapa presentaba la peculiaridad de que los registros recabados no se encontraban presentados por película, sino como un conjunto de *tweets* para cada película, por lo que se debió realizar de tal manera que posteriormente se generara un conjunto de datos con registros para cada una de las películas pertenecientes al estudio. Para ello los archivos se modificaron de la siguiente manera (véase tablas 13 y 14 e imagen 9 y 10):

- ***Tweets* de las cuentas oficiales:**

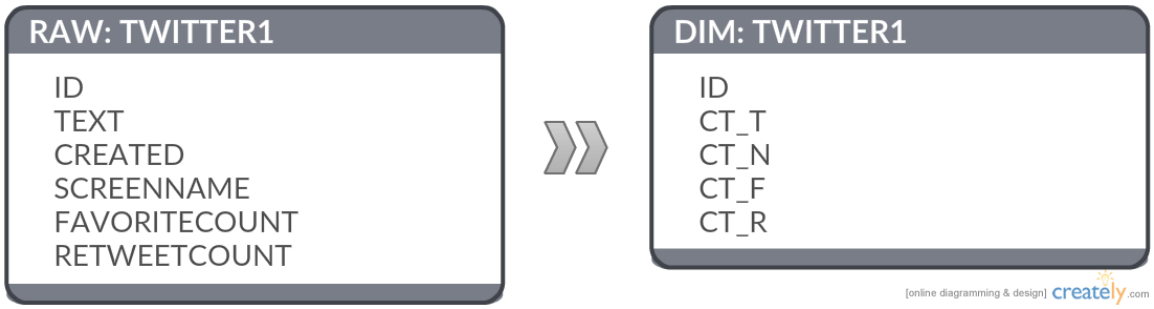
Tabla 13: Preparación de las variables extraídas de las cuentas oficiales de promoción de las películas en Twitter.

Variable Origen		*Variable Nueva*	*Descripción*
TEXT		***TEXT***	Texto del *tweet*. Variable eliminada, pero utilizada para contar el número de *tweets* generados
		CT_N	Número de *tweets* publicados por la cuenta
CREATED		***CREATED***	Fecha de creación del *tweet*
SCREENNAME		***SCREENNAME***	Nombre de la cuenta que generó el *tweet*. Variable eliminada, pero utilizada para determinar el tipo de cuenta
		CT_T	Tipo de cuenta Twitter. Codificada de manera nominal. • ***-1:*** Cuenta corporativa (estudio) • ***0:*** No tiene cuenta • ***1:*** Cuenta propia de la película
FAVORITECOUNT		***CT_F***	Cantidad de favoritos totales de la cuenta para la fecha de estudio. Variable generada mediante la suma de los favoritos de cada *tweet* individual
RETWEETCOUNT		***CT_R***	Cantidad de *retweets* totales de la cuenta para la fecha de estudio

Variable generada mediante la suma de los *retweets* de cada *tweet* individual

No se modificó | Variable generada a partir de la original | Variable eliminada

Imagen 9: Formato de la tabla de cuentas oficiales de Twitter en la base de datos



- ***Tweets* de otros usuarios**

Tabla 14: Preparación de las variables de otros usuarios de Twitter.

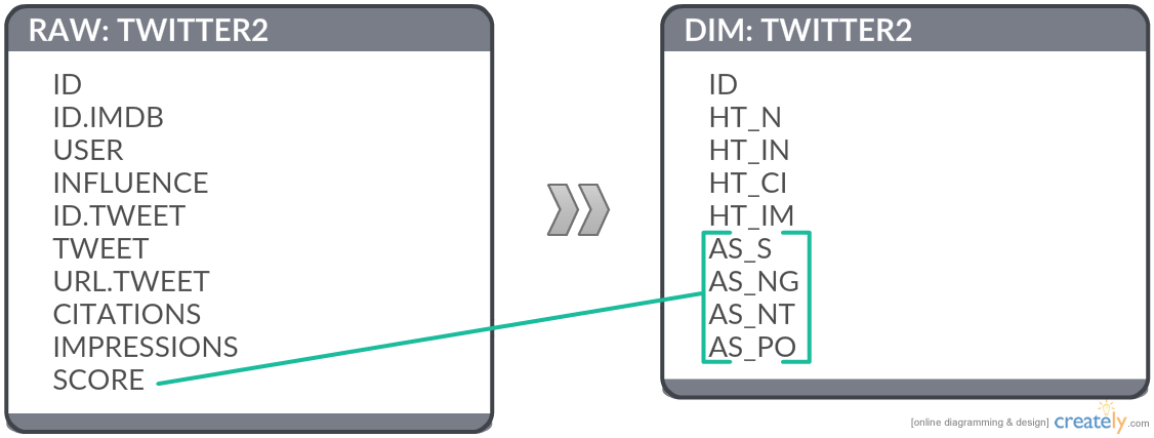
Variable Origen	*Variable Nueva*	*Descripción*
ID	***ID***	Número de control del *tweet*
	HT_N	Número de t*weets* obtenidos de Topsy del *hashtag* de la película entre 30 y 10 días antes del estreno, con un máximo de 401
ID.IMDB	***ID.IMDB***	ID según IMDB de la película a la que se refiere el *tweet*
USER	***USER***	Usuario que generó el *tweet*
INFLUENCE	***HT_IN***	Nivel promedio de influencia del total de *tweets* referidos a una misma película Variable calculada como el promedio de influencia de los autores de los *tweets*

ID.TWEET		***ID.TWEET***	ID de control del *tweet* de acuerdo a Twitter.
TWEET		***TWEET***	Texto del *tweet* Variable eliminada, pero utilizada para el análisis de sentimientos
URL.TWEET		***URL.TWEET***	URL del *tweet*
CITATIONS		***HT_CI***	Nivel promedio de citas de los *tweets* referidos a una misma película Variable calculada como el promedio de citaciones de los *tweets*
IMPRESSIONS		***HT_IM***	Nivel promedio de impresiones de los *tweets* referidos a una misma película. Variable calculada como el promedio de impresiones de los *tweets*
SCORE		***SCORE***	Valor de polaridad para cada *tweet* Variable eliminada, pero utilizada para generar las siguientes 4 variables
		AS_S	Valor promedio del sentimiento de los t*weets* referidos a una misma película
		AS_NG	Porcentaje de t*weets* negativos publicados utilizando el *hashtag* de la película
		AS_NT	Porcentaje de t*weets* neutros publicados utilizando el *hashtag* de la película

	AS_PO	Porcentaje de *tweets* positivos publicados utilizando el *hashtag* de la película

No se modificó | Variable generada a partir de la original | Variable eliminada

Imagen 10: Tabla de información de Twitter para cuentas de usuarios en la base de datos.



d. YouTube

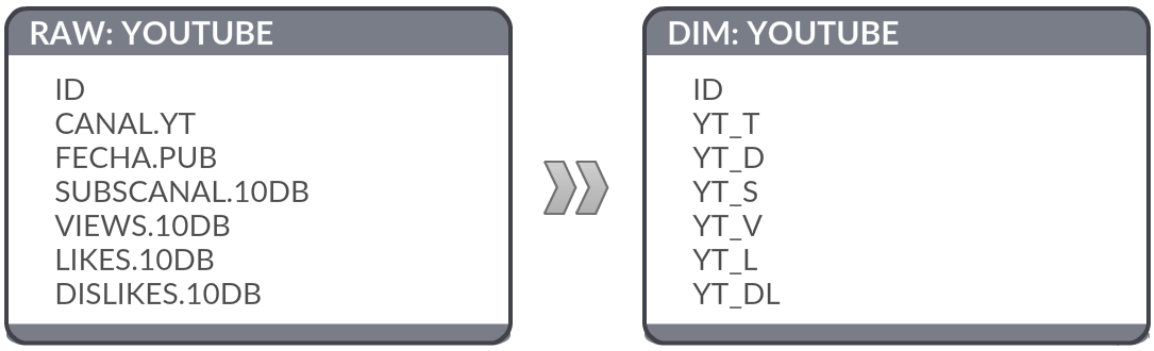
Tabla 15: Preparación de las variables extraídas de YouTube.

Variable Origen	*Variable Nueva*	*Descripción*
CANAL.YT	***CANAL.YT***	Canal que publicó el video
	YT_T	Tipo de canal de YouTube que ha logrado el mayor número de visitas en el tráiler oficial de la película. Esta variable se codificó de manera nominal, siendo: • **1:** Canal público • **2:** Canal Oficial de la película • **3:** Canal corporativo (estudio)
FECHA.PUB	***FECHA.PUB***	Fecha de publicación del vídeo. Variable eliminada, pero utilizada para calcular la variable "***YT_D***"
	YT_D	Número de días que el vídeo lleva subido antes del estreno (hasta 10 días antes del

			estreno)
SUBSCANAL.10DB		***YT_S***	Número de suscriptores del canal con mayor número de visitas (hasta 10 días antes del estreno)
VIEWS.10DB		***YT_V***	Número de visualizaciones del tráiler oficial de la película tiene (hasta 10 días antes del estreno)
LIKES.10DB		***YT_L***	Número de *likes* que el tráiler oficial de la película tiene hasta 10 días antes del estreno
DISLIKES.10DB		***YT_DL***	Número de *dislikes* que el tráiler oficial de la película tiene hasta 10 días antes del estreno

No se modificó | Variable generada a partir de la original | Variable eliminada

Imagen 11: Formato de tabla de YouTube a incluir en la base de datos.



4 CREACIÓN DE LA BASE DE DATOS

4.1 CREACIÓN DE LA BASE DE DATOS

Para el desarrollo del trabajo decidimos crear una base de datos que nos permitiera almacenar tanto los datos crudos, como los datos limpios, con la intención de garantizar la integridad de los mismos, y permitir un mejor manejo de altos volúmenes de información.

En este propósito seleccionamos MS SQL SERVER, una base datos relacional, que utiliza como lenguaje Transact- SQL, y que justifica su uso en la capacidad de manejo y

explotación de registros transaccionales (aspecto en el que aventajan a las bases de datos NoSQL, las cuales son más convenientes para el manejo de datos no estructurados).

Para esto creamos dos instancias de almacenamiento (véase imagen 12), una temporal para datos crudos, y otra final para datos procesados y limpios, a partir de los cuales creamos nuestra tabla maestra, la cual funcionará como fuente principal para futuros estudios.

Imagen 12: Esquema general de relación base de datos.

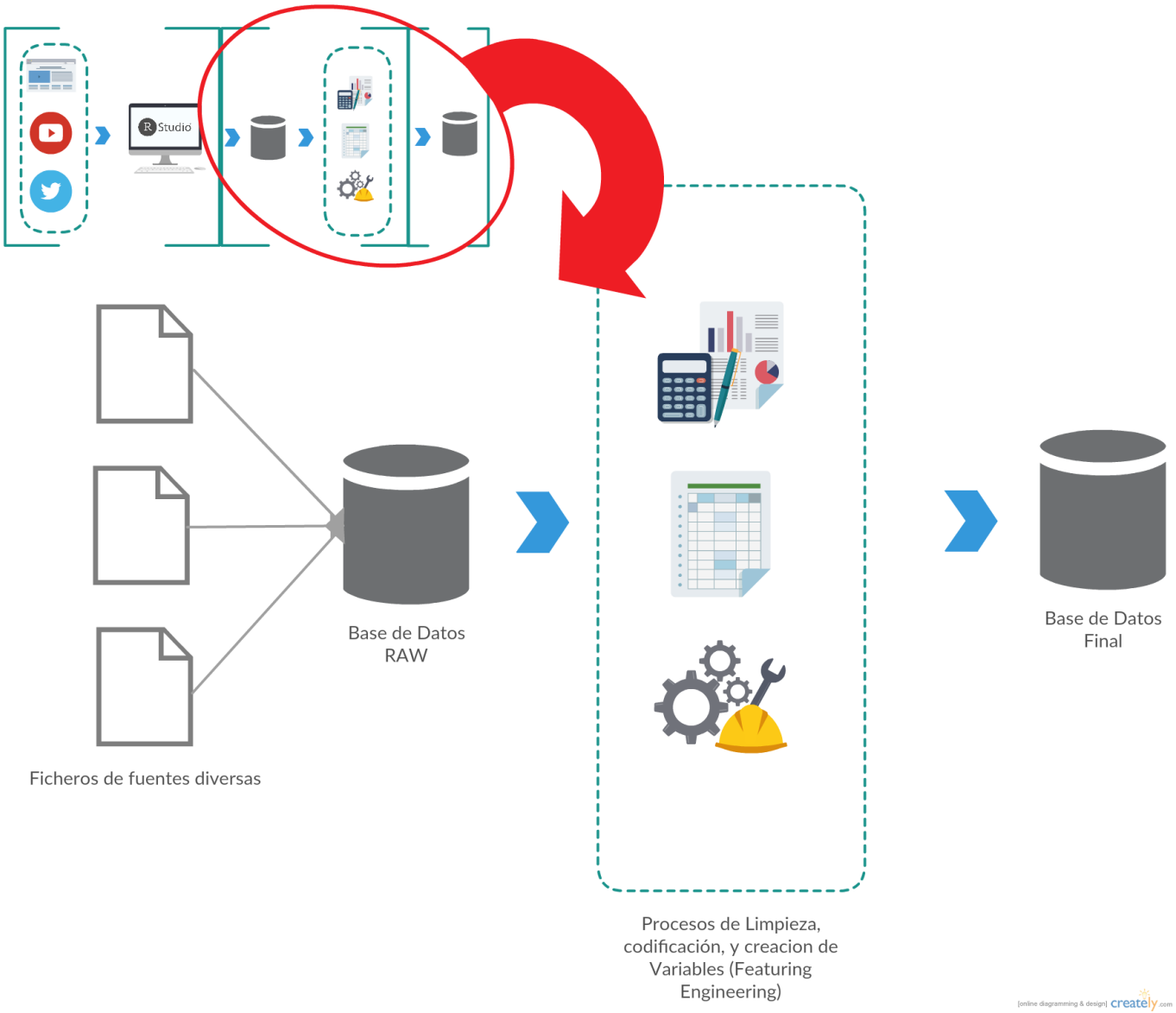
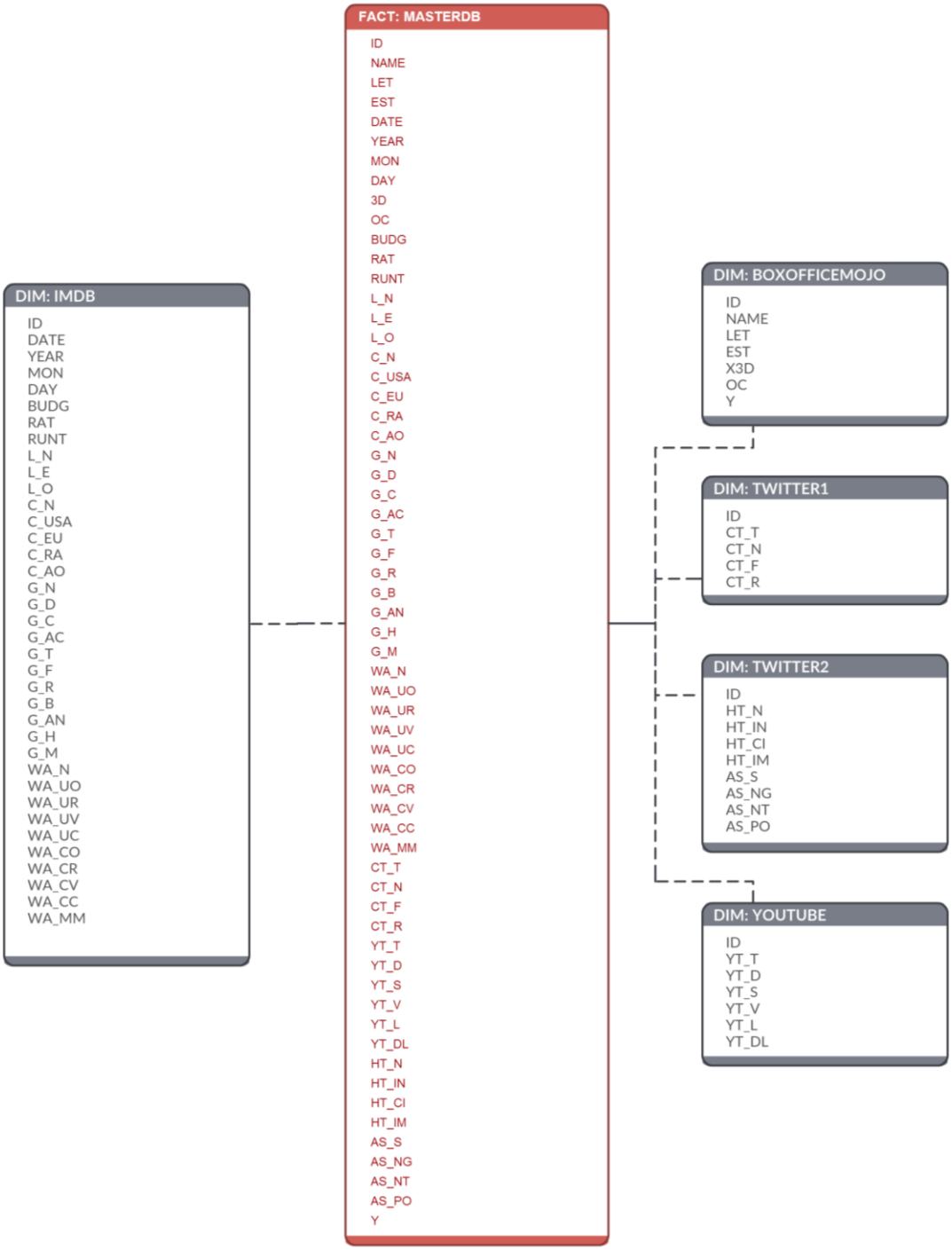


Imagen 13: Relación y dimensiones de las tablas de hechos.



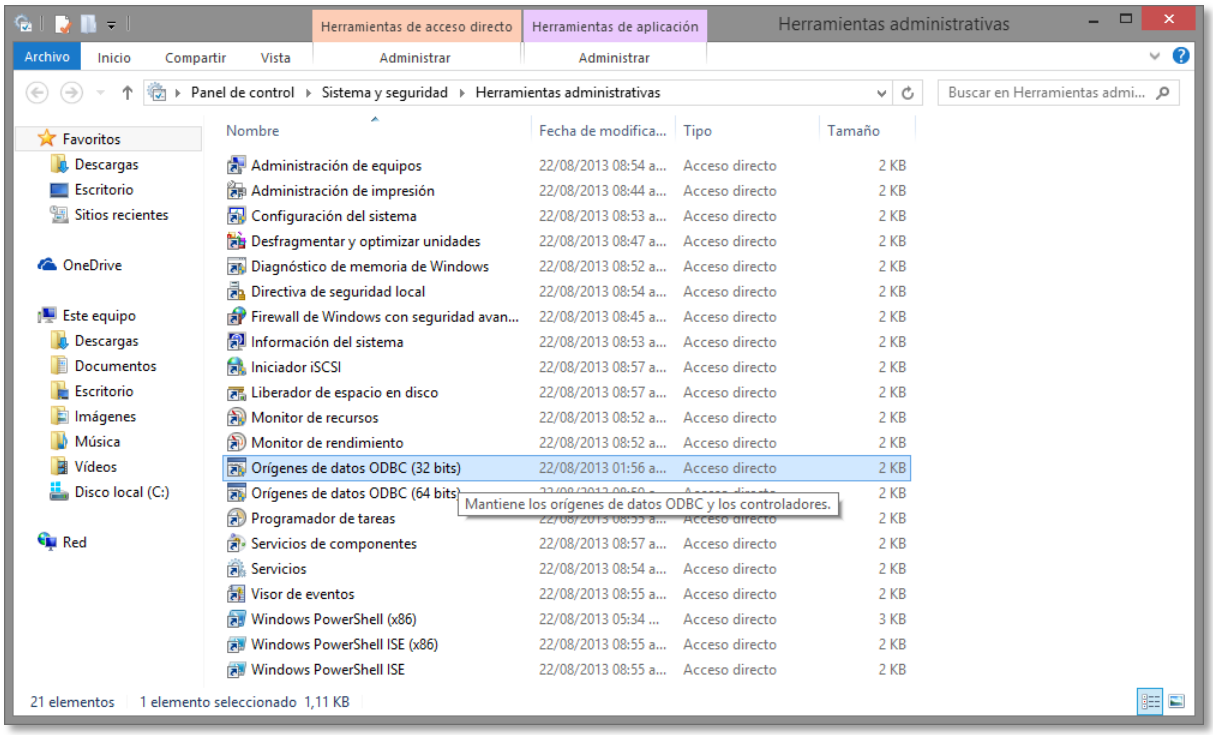
4.2 CREACIÓN DE CONEXIÓN ODBC

Para poder utilizar nuestra base de datos como fuente de información debemos crear una capa ODBC (*Open Database Connectivity*), capaz de conectar, mediante una capa

intermedia, la aplicación que estamos utilizando con el sistema de gestión de base de datos (DBMS: *Database Management System*). El propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el DBMS entienda.

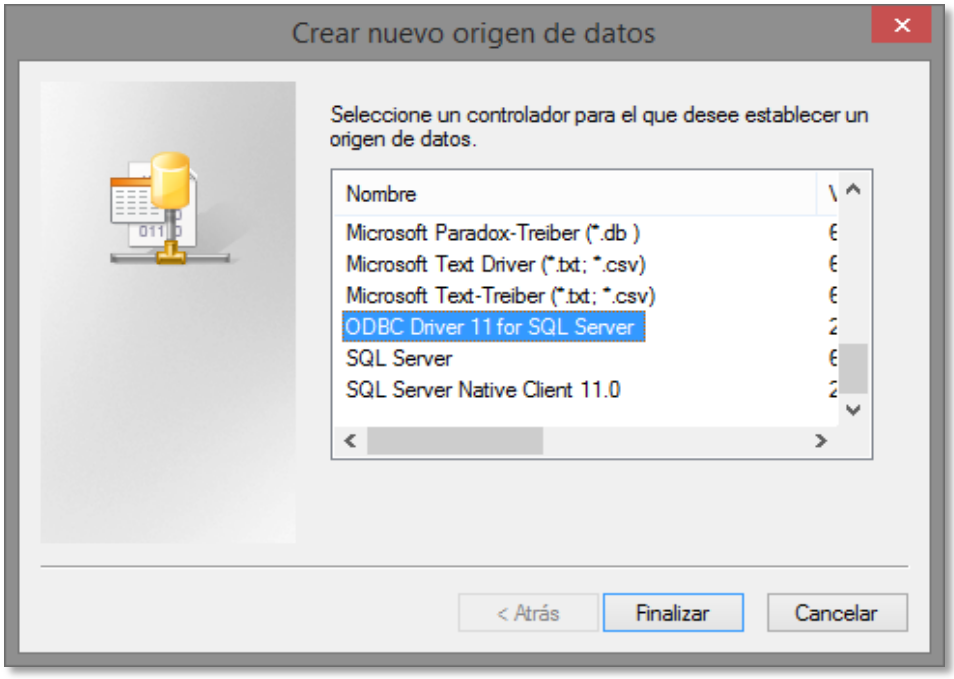
Esto se realiza desde la opción de Herramientas administrativas, ubicada en el panel de control del equipo donde se localiza la base de datos (véase imagen 14).

Imagen 14: Creación de la capa ODBC



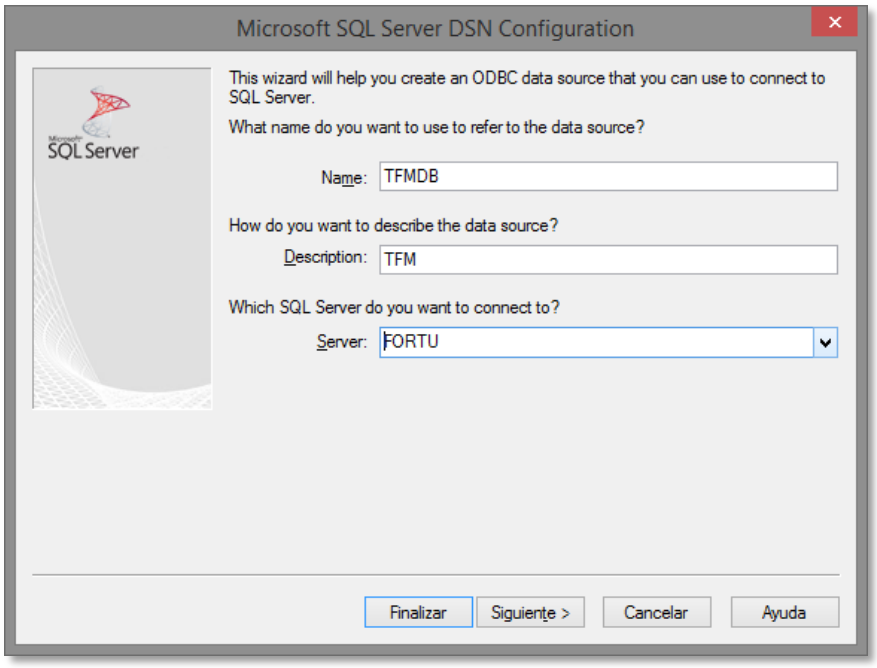
Al seleccionar la opción “Agregar”, se desplegará un menú con los diferentes controladores para conexión. En nuestro caso la opción “ODBC Driver 11 for SQL Server” (véase imagen 15).

Imagen 15: Elección del controlador para conexión.



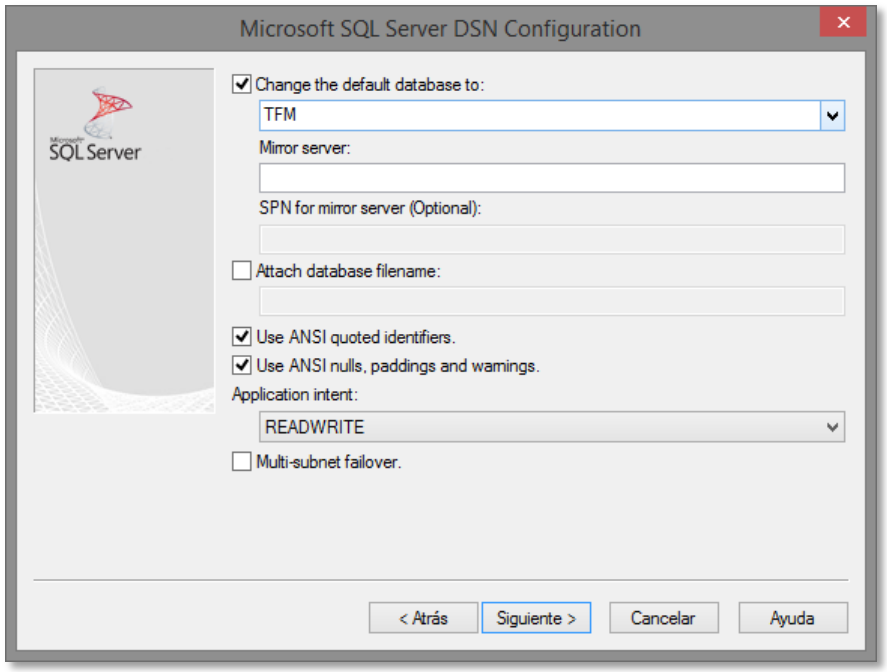
Luego añadimos un nombre y una descripción a nuestra capa de conexión, y seleccionamos el nombre del servidor al que deseamos conectarnos por medio de dicha capa (éste deberá estar previamente creado dentro del SQL Server) (véase imagen 16).

Imagen 16: Asignación de un identificador para la conexión con la Base de Datos.



Finalmente, seleccionamos cual será la base de datos por defecto a la que se conectará la capa para las acciones de lectura y escritura (esta debe ser creada previamente dentro del servidor que mencionamos en el paso anterior) (véase imagen 17).

Imagen 17: Selección de la base de datos por defecto



4.3 CARGA DE DATOS EN LA BASE DE DATOS: CONEXIÓN CON R.

Para la carga de datos en la base de datos utilizamos el paquete "RODBC"[14], el cual permite a RStudio conectarse con la capa creada anteriormente utilizando el controlador ODBC (véase código 12).

Código 12: Librería para conectar la base de datos con RStudio

```
install.packages("RODBC")
library(RODBC)
```

Este paquete permite leer, crear y eliminar tablas en la base de datos seleccionada utilizando las siguientes funciones principales (véase tabla 16):

Tabla 16: Principales Funcionalidades del paquete RODBC.

odbcConnect()	Establece la conexión a la base de datos

[14] https://cran.r-project.org/web/packages/RODBC/RODBC.pdf

sqlFetch()	Lee una tabla mediante una capa ODBC y la importa a RStudio como un *dataframe*
sqlQuery()	Envía una consulta (utilizando lenguaje SQL) y regresa los resultados en un *dataframe,* mediante el uso del comando sqlGetResults()
sqlSave()	Escribe o actualiza una tabla de la base de datos mediante la interfaz ODBC
sqlDrop()	Elimina una tabla de la base de datos
close()	Cierra la conexión con la base de datos

Seguidamente debe crearse la conexión con la capa ODBC mediante el Sistema de Nombres de Dominio (DNS: *Domain Name System*) asignado, y realizar la operación que necesitemos (véase código 13).

Código 13: Conexión con la base de datos

```
conex <-odbcConnect("TFMDB")
sqlSave(conex, masterdb, tablename = "masterdb", append = FALSE)
odbcClose(conex)
```

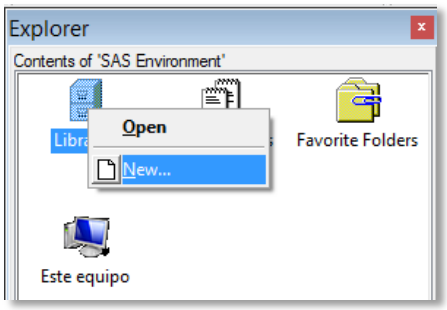
Y de esta manera se crea la tabla maestra en la base de datos, utilizando la capa ODBC, a partir de un *dataframe* de RStudio.

4.4 CONEXIÓN CON SAS

Una vez creada la base de datos a partir de SQL Server y alimentada con la información de las redes sociales que tenemos preparada, debe crearse primeramente una librería asociada el en entorno SAS. Esta tarea se realizó con la finalidad de dar continuidad al estudio posterior de los datos, los cuales se realizarán con apoyo del software SAS Base.

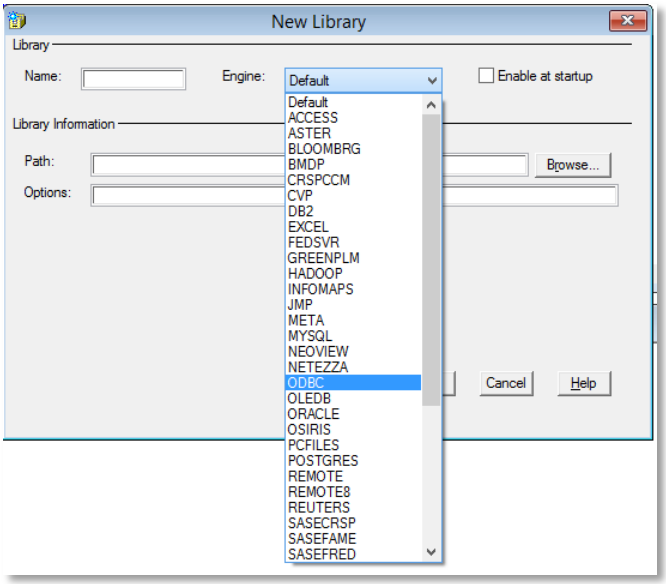
Para esto, hacemos clic derecho sobre el icono de librería, y luego presionamos en "New" para iniciar el asistente de creación de librerías (véase imagen 18).

Imagen 18: Creación de la librería en SAS Base



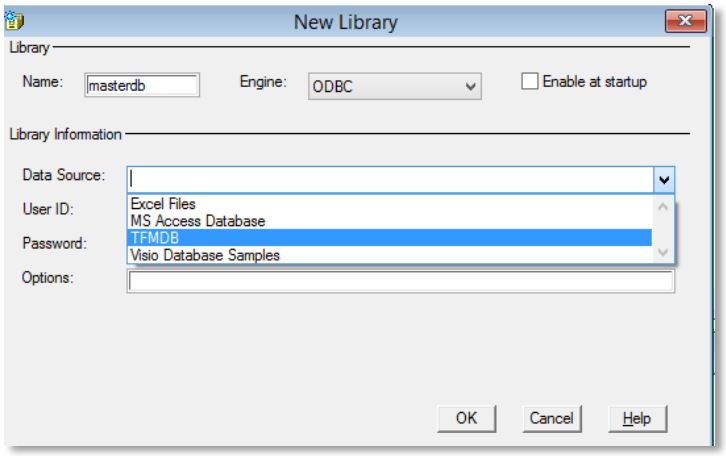
Dentro del asistente, desplegamos el combo de opciones “Engine” y seleccionamos la opción “ODBC”, correspondiente a la capa que hemos creado anteriormente (véase imagen 19).

Imagen 19: Conexión con la base de datos



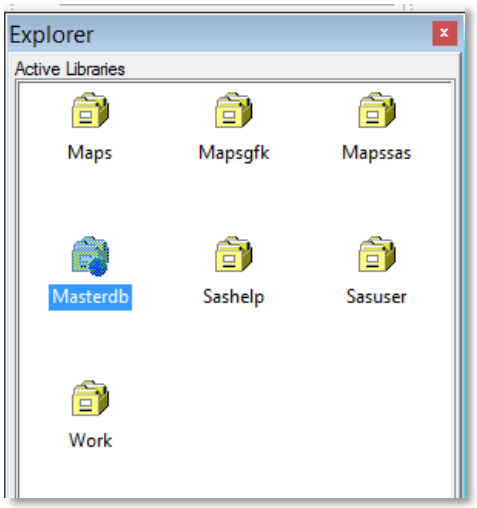
Luego, procedemos a nombrar esta nueva librería, y en la casilla “Data Source” seleccionamos la capa “ODBC” creada previamente (véase imagen 20).

Imagen 20: Creación de la base de datos



Al presionar “OK”, aparecerá la nueva librería en nuestra lista, y todas las tablas creadas dentro de la base de datos aparecerán como fuentes de datos SAS listas para su uso.

Imagen 21: Librería con conexión a la base de datos



5 CONCLUSIONES

"The ability to take data—to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it—that's going to be a hugely important skill in the next decades, not only at the professional level but even at the educational level for elementary school kids, for high school kids, for college kids. Because now we really do have essentially free and ubiquitous data. So the complimentary scarce factor is the ability to understand that data and extract value from it."

(Varian, 2009)

Las técnicas y herramientas informáticas, son recursos muy valiosos (incluso necesarios) para el desarrollo de proyectos de minería de datos. En una era en la que la generación de datos crece a pasos vertiginosos, tanto en volumen como en diversidad, saber aprovechar estas fuentes definitivamente representa una ventaja competitiva y altamente beneficiosa en la investigación y generación de conocimientos.

Como puntos concluyentes, podemos aportar lo siguiente:

- El manejo de herramientas informáticas (técnicas de extracción y limpieza de datos, dominio de lenguajes de programación, manejo de bases de datos) son fundamentales en las actividades de recopilación de datos, y pueden representar una gran diferencia al momento de la ejecución de un proyecto de investigación.
- La web (incluyendo redes sociales) puede ser una gran fuente de conocimiento, si sabemos extraer información de los datos que nos aporta. En este punto, una adecuada ejecución de técnicas como la extracción y la ingeniería de características (Feature Extraction y Feature Engineering) pueden ayudarnos a disminuir la incertidumbre y a aportar relevancia, propósito y contexto a nuestros datos.
- A pesar de lo dicho en el párrafo anterior, debemos tener en cuenta que mucha de la información en la web presenta limitaciones, como la temporalidad de los datos, en donde la información solo es relevante al momento de su consulta, lo

cual dificulta su consulta en estados históricos o pasados. Pocas fuentes permiten consultar evoluciones históricas de un mismo dato. En nuestro caso en particular, nos creó los siguientes inconvenientes:

  - Nos obligó a descartar algunas variables, que considerábamos valiosas, como la presencia de determinados actores o quien fue el director de la película. Estas variables pueden influir en el impacto que la misma creará en los consumidores, pero la ausencia de datos históricos relacionados a los artistas involucrados nos impidió determinar el nivel de influencia que pudieron tener los mismos al momento del estreno de la película.
  - En la actividad de extracción de código HTML de YouTube (*scraping*), considerando que estábamos recopilando información de hasta 3 años de antigüedad, se nos presentaron, en repetidas ocasiones, cambios en la estructura de la página, lo que nos obligó a supervisar y reajustar reiteradamente el código para poder completar la tarea de manera satisfactoria.
  - Fuentes valiosas, como Twitter, solo permiten, por medio de su API gratuita, la extracción de *tweets* como no más de 7 días de antigüedad, lo que nos obligó a descartar este proveedor, y tener que contactar con un proveedor de pago para obtener dicha información.

- Complementado lo comentado, debemos exponer que, aunque existe mucha información "libre y pública", también existe una parte de la información (muy valiosa, ya que principalmente se trata de información muy enriquecida y especializada) que sólo puede ser extraída mediante empresas proveedoras de datos que suelen requerir pago por sus servicios, y que debemos considerar al momento de determinar las fuentes que nos proporcionarán los datos al inicio del proyecto.
- Es necesario planificar claramente las actividades de extracción de datos, ya que pueden existir fuentes que establezcan limitaciones de descarga que pueden afectar al adecuado desarrollo del proyecto.

- Otro punto clave en las actividades de generación de datos es la homogeneidad de los mismos. En nuestro caso, cuidar el factor temporal en los datos recolectados es vital para evitar incongruencias en los mismos.

5.1 TRABAJOS FUTUROS

El presente trabajo se ha realizado como proceso inicial de un proyecto de investigación sobre el sector cinematográfico, y servirá de fase previa para el desarrollo del mismo. En él, se desarrollarán modelos de predicción con la intención de encontrar un modelo que permita anticipar los ingresos a obtener por una película en su fin de semana de estreno.

Como parte de puntos de desarrollos adicionales, se propone para trabajos futuros, la adición de otra fuente de datos (Facebook), la cual podría aportar datos complementarios y muy interesantes.

6 BIBLIOGRAFÍA

Bowles, Michael. 2015. *Machine Learning in Python: Essential Techniques for Predictive Analysis.* Indianapolis : John Wiley & Sons, 2015. ISBN: 1118961765.

Guyon, Isabelle, y otros. 2008. *Feature Extraction: Foundations and Applications.* California : Springer, 2008. ISBN: 3540354883.

Liu, Bing, Hu, Minqing y Cheng, Junsheng. 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web. Chiba, Japón : 14th International World Wide Web conference, 2005.

Munzert, Simon, y otros. 2015. *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining.* West Sussex : John Wiley & Sons, 2015. ISBN: 111883481X.

O'Neil, Cathy y Schutt, Rachel. 2013. *Doing Data Science: Straight Talk from the Frontline.* California : O'Reilly Media, Inc., 2013. ISBN: 144936389X.

Varian, Hal. 2009. How the Web challenges managers. *McKinsey & Company.* [En línea] Enero de 2009. http://www.mckinsey.com/insights/innovation/hal_varian_on_how_the_web_challenges_managers.

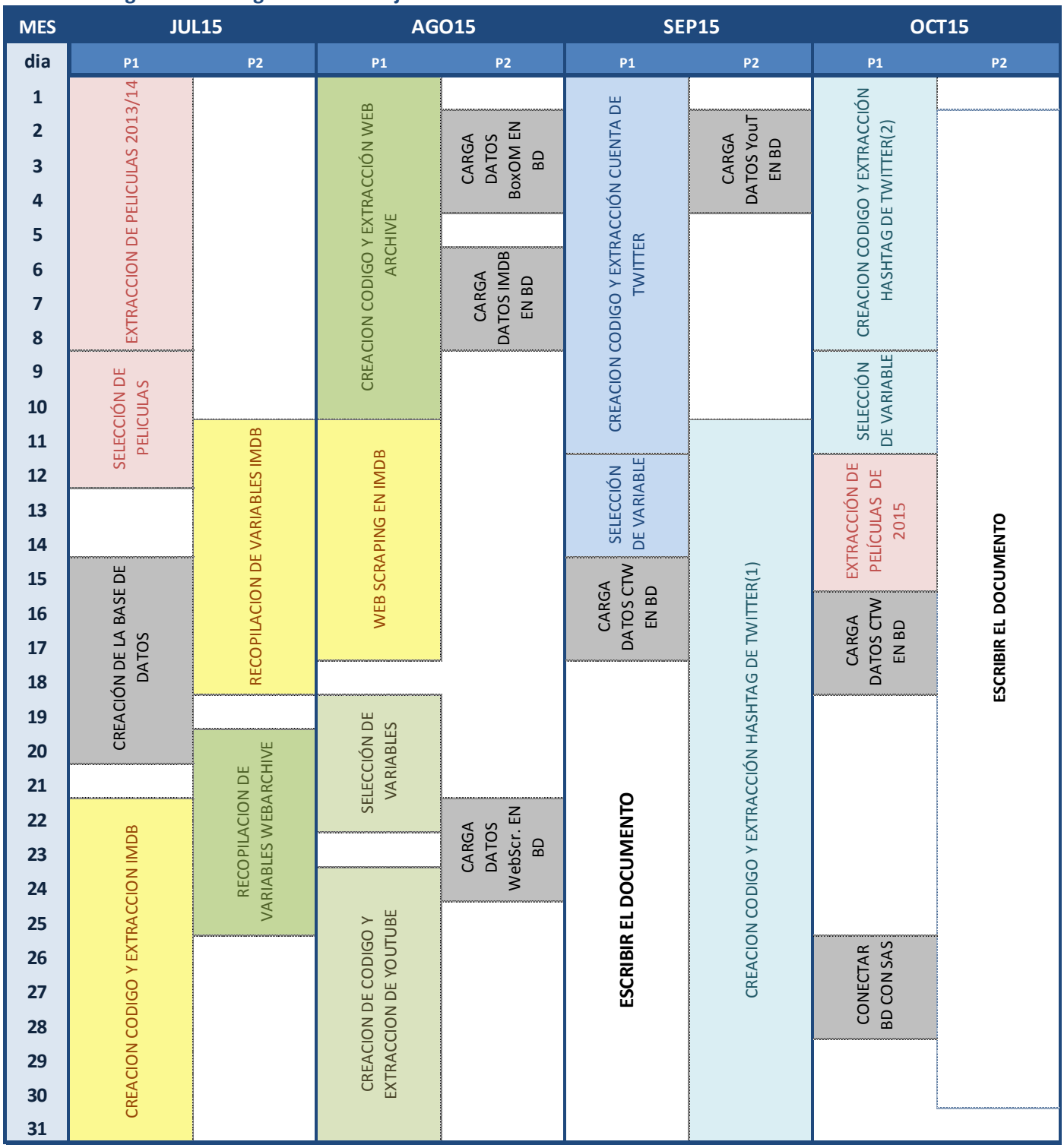
7 ANEXOS.

7.1 CRONOGRAMA ESQUEMAD DE TRABAJO

LEYENDA:

BoxOfficeMojo	IMDB	WebArchive	YouTube	Twitter Cuenta	Twitter Hashtag	Base Datos

Imagen 22: Cronograma de trabajo diario*



***todas las fechas son aproximadas**

Se muestra en la tabla (véase imagen 22) el esquema de trabajo seguido para desarrollar todo el proceso llevado a cabo en la extracción de la información. Indicar que parte del trabajo se llevó a cabo simultáneamente, de ahí P1 (Proceso principal) y P2 (Proceso secundario) de cada día de trabajo.

Además de lo expuesto, el mes de Junio nos sirvió como período de recabar información y armonizar hasta qué punto podíamos abarcar con el alcance del trabajo.

Los mayores contratiempos vieneron de Twitter y YouTube:

- **Twitter**: El trabajar con información tan específica de twitter (fechas anteriores a 7 días) sin contar con cuentas profesionales -y de pago-. Nos llevó a afrontar dos limitaciones: La primera vino a la hora de extraer información de la cuenta oficial, puesto que teníamos una limitación de extraer 3200 tweets por hora, lo que nos llevó a tardar 9 días en obtener los 220.000 tweets con los que contamos a pesar de trabajar más de 8 horas al día. La segunda provino de la extracción de los tweets del término de búsqueda, en este caso la limitación era de 7000 tweets al día, lo que nos llevó a tardar más de 20 días para obtener los 160.000 tweets con los que contamos en este apartado.
- **YouTube**: Esta web tenía un código con alta requerimiento computacional, lo que unido a la limitación de que el código html cambiaba continuamente a lo largo del tiempo; nos obligaba a lanzar los códigos varias veces durante varios días para obtener toda la información necesaria.